PROFEAT 2016

Introduction of Descriptors

Table of Contents

In the following sections, we will illustrate the protein-related descriptors calculated by PROFEAT (2016) in detail, including protein and peptide, small molecules, protein-protein interaction pair, protein-ligand interaction pair, and protein-protein network.

1. Protein and Peptide Descriptors

A protein or peptide sequence with N amino acid residues is expressed as:

R_1, R_2, R_3…R_N, where R_i represents the residue at the i-th position in the sequence. The labels i and j are used to index amino acid position in a sequence and r, s are used to index the amino acid type. The computed features are divided into 4 groups according to their known applications described in the literature. A protein sequence can be divided equally into segments and the methods, described as follows for the global sequence, can be applied to each segment.

1.1 Feature Group 1 [G1]: Amino acid composition

The amino acid composition is the fraction of each amino acid type within a protein. The fractions of all 20 natural amino acids are calculated as:

$$f(r) = \frac{N_r}{N} \qquad r = 1, 2, 3 \ldots 20 \tag{1}$$

Where N_r is the number of the amino acid type r and N is the length of the sequence.

1.2 Feature Group 2 [G2]: Dipeptide composition

The dipeptide composition gives 400 features, defined as:

$$fr(r,s) = \frac{N_{rs}}{N-1} \qquad r, s = 1,2,3,\ldots,20 \tag{2}$$

Where N_{ij} is the number of dipeptide represented by amino acid type r and s.

Reference:

[1] M. Bhasin and G. P. S. Raghava. Classification of Nuclear Receptors Based on Amino Acid Composition and Dipeptide Composition. J. Bio. Chem. 2004, 279, 23262.

1.3 Feature Group 3 [G3]: Autocorrelation descriptors

Autocorrelation descriptors are defined based on the distribution of amino acid properties along the sequence. The amino acid properties used here are various types of amino acids index (http://www.genome.ad.jp/dbget/aaindex.html).Three type of autocorrelation descriptors are used here and are described as following.

All the amino acid indices are centralized and standardized before the calculation, i.e.

$$P_r^{'} = \frac{P_r - \overline{P}}{\sigma} \quad (3)$$

Where $\overline{P}$ is the average of the property of the 20 amino acids.

$$\overline{P} = \frac{\sum_{r=1}^{20} P_r}{20} \quad (4)$$

And

$$\sigma = \sqrt{\frac{1}{20}\sum_{r=1}^{20}(P_r - \overline{P})^2} \quad (5)$$

1.3.1 Normalized moreau-broto autocorrelation descriptors

Moreau-Broto autocorrelation descriptors application to protein sequences may be defined as:

$$AC(d) = \sum_{i=1}^{N-d} P_i P_{i+d} \qquad \text{d=1, 2, 3 … nlag} \quad (6)$$

Where d is called the lag of the autocorrelation and P_i and P_{i+d} are the properties of the amino acids at position i and i+d , respectively. nlag is the maximum value of the lag.

The normalized Moreau-Broto autocorrelation descriptors are defined as:

$$ATS(d) = \frac{AC(d)}{N-d} \qquad \text{d=1, 2, 3 … nlag} \quad (7)$$

1.3.2 Moran autocorrelation

Moran autocorrelation descriptors application to protein sequence may be defined as:

$$I(d) = \frac{\frac{1}{N-d}\sum_{i=1}^{N-d}(P_i - \overline{P})(P_{i+d} - \overline{P})}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P})^2} \qquad \text{d=1, 2, 3 … 30.} \quad (8)$$

Where d and P_i and P_{i+d} are defined in the same way as in 2.2.1, and $\overline{P}$ is the average of the considered property P along the sequence, i.e.

$$\overline{P} = \frac{\sum_{i=1}^{N} P_i}{N} \quad (9)$$

Where d, $\overline{P}$, P_i and P_{i+d} , nlag have the same meaning as in the above.

1.3.3 Geary autocorrelation Descriptors

Geary autocorrelation descriptors application to protein sequence may be defined as:

$$C(d)=\frac{\frac{1}{2(N-d)}\sum_{i=1}^{N-d}(P_i - P_{i+d})^2}{\frac{1}{N-1}\sum_{i=1}^{N}(P_i - \overline{P}\ )^2} \qquad \text{d=1, 2, 3 … 30.} \qquad (10)$$

Where d, $\overline{P}$, P_i and P_{i+d} , nlag have the same meaning as in the above.

The amino acid indices used in these auto-correlation descriptors can be specified in file "input-param.dat" from "input-aaindexdb.dat".

For each amino acid index, there will be 3×nlag auto-correlation descriptors.

1.4 Feature Group 4 [G4]: Composition, transition and distribution

These descriptors are developed by Dubchak, et.al.

Step1. Sequence encoding

The amino acids are divided in three classes according to its attribute and each amino acid is encoded by one of the indices 1, 2, 3 according to which class it belonged.

The attributes used here include hydrophobicity, normalized van der Waals volume, polarity, and polarizability, as in the references. The corresponding division is in the table 4.

Table 4 Amino Acid attributes and the Division of the Amino Acids

ID	Property	Class 1	Class 2	Class 3
[G4.x.1]	Hydrophobicity	Polar RKEDQN	Neutral GASTPHY	Hydrophobicity CLVIMFW
[G4.x.2]	Normalized van der Waals volume	0-2.78 GASTPD	2.95-4.0 NVEQIL	4.03-8.08 MHKFRYW
[G4.x.3]	Polarity	4.9-6.2 LIFWCMVY	8.0-9.2 PATGS	10.4-13.0 HQRKNED
[G4.x.4]	Polarizability	0-1.08 GASDT	0.128-0.186 CPNVEQIL	0.219-0.409 KMHFRYW
[G4.x.5]	Charge	Positive KR	Neutral ANCQGHILMFPSTWYV	Negative DE
[G4.x.6]	Secondary structure	Helix EALMQKRH	Strand VIYCWFT	Coil GNPSD
[G4.x.7]	Solvent accessibility	Buried ALFCGIVW	Exposed PKQEND	Intermediate MPSTHY
[G4.x.8]	Surface tension	-0.20~0.16 GQDNAHR	-0.3~ -0.52 KTSEC	-0.98~ -2.46 ILMFPWYV
[G4.x.9]	Protein-protein interface hotspot propensity - Bogan	High (5-21%) DHIKNPRWY	Medium (1.12-3.64%) EQSTGAMF	Low (0-0.83%) CLV
[G4.x.10]	Protein-protein interface propensity - Ma	High (1.21-2.02) CDFMPQRWY	Medium (0.63-1.12) AGHVLNST	Low (0.14-0.29) EIK

[G4.x.11]	Protein-DNA interface propensity - Schneider	High (4-30%) GKNQRSTY	Medium (1-3%) ADEFHILVW	Low (0-1%) CMP
[G4.x.12]	Protein-DNA interface propensity - Ahmad	High (25-100%) GHKNQRSTY	Medium (5-18%) ADEFIPVW	Low (0-4%) CLM
[G4.x.13]	Protein-RNA interface propensity - Kim	High (0.25-11) HKMRY	Medium (-0.25 – 0.17) FGILNPQSVW	Low (-0.3 - -0.8) CDEAT
[G4.x.14]	Protein-RNA interface propensity - Ellis	High (1.18-2.07) HGKMRSYW	Medium (0.84-1.16) AFINPQT	Low (0.41-0.8) CDELV
[G4.x.15]	Protein-RNA interface propensity - Phipps	High (0.95-1.8) HKMQRS	Medium (0.5-0.95) ADEFGLNPVY	Low (0-0.5) CITW
[G4.x.16]	Protein-ligand binding site propensity - Khazanov	High (⩾2.25) CFHWY	Medium (1.6-2.3) GILNMSTR	Low (⩽1.5) AEDKPQV
[G4.x.17]	Protein-ligand valid binding site propensity - Khazanov	High (⩾1.4) CFHWYM	Medium (0.79-1.21) DGILNSTV	Low (⩽0.76) AEKPQR
[G4.x.18]	Propensity for protein-ligand polar & aromatic non-bonded interactions - Imai	High (477-1197) DEHRY	Medium (95-423) CFKMNQSTW	Low (<95) AGILPV
[G4.x.19]	Molecular Weight	Low (75-105) AGS	Medium (115-155) CDEHIKLMNQPTV	High (165-204) FRWY
[G4.x.20]	cLogP	-4.2 - -3.3 RKDNEQH	-3.07 – 2.26 PYSTGACV	-1.78 - -1.05 WMFLI
[G4.x.21]	No of hydrogen bond donor in side chain	>1 HKNQR	1 DESTWY	0 ACGFILMPV
[G4.x.22]	No of hydrogen bond acceptor in side chain	>1 DEHNQR	1 KSTWY	0 ACGFILMPV
[G4.x.23]	Solubility in water	High (9-65 g/100g) ACGKRT	Medium (1.14-7.44 g/100g) EFHILMNPQSVW	Low (0.048-0.82 g/100g) DY
[G4.x.24]	Amino acid flexibility index	Very flexible EGKNQS	Moderately flexible ADHIPRTV	Less flexible CFLMWY

For example, for a given sequence “MTEITAAMVKELRESTGAGA”, it will be encoded as “32132223311311222222” according to its hydrophobicity division.

Step 2: Composition, Transition and Distribution descriptors

The ‘x’ in descriptor ID in table 4 can be either ‘1’ or ‘2’ or ‘3’, which represents three different feature categories 1: “Composition (C)”, 2: “Transition (T)”, and 3: “Distribution (D)” respectively. Their calculation details for a given attribute are as follows:

Composition: It is the global percent for each encoded class in the sequence. In the above example using Hydrophobicity division, the numbers for encoded classes “1”, “2”, “3” are 5, 10, 5 respectively, so the compositions for them are 5/20=25%, 10/20=50%, and 5/20=25% respectively, where 20 is the length of the protein sequence. Composition can be defined as:

$$C_r = \frac{n_r}{N} \qquad r=1, 2, 3 \tag{11}$$

Where n_r is the number of r in the encoded sequence and N is the length of the sequence.

Transition: A transition from class 1 to 2 is the percent frequency with which 1 is followed by 2 or 2 is followed by 1 in the encoded sequence. Transition descriptor can be calculated as:

$$T_{rs} = \frac{n_{rs} + n_{sr}}{N-1} \qquad rs = \text{“12”, “13”, “23”} \tag{12}$$

Where n_{rs}, n_{sr} is the numbers of dipeptide encoded as “rs” and “sr” respectively in the sequence and N is the length of the sequence.

Distribution: The “distribution” descriptor describes the distribution of each attribute in the sequence. There are five “distribution” descriptors for each attribute and they are the position percents in the whole sequence for the first residue, 25% residues, 50% residues, 75% residues and 100% residues , respectively, for a specified encoded class. For example, there are10 residues encoded as “2” in the above example, the positions for the first residue “2”, the 2th residue “2” (25%*10=2), the 5th “2” residue (50%*10=5), the 7th “2” (75%*10=7) and the 10th residue “2” (100%*10) in the encoded sequence are 2, 5, 15, 17,20 respectively, so the distribution descriptors for “2” are: 10.0 (2/20*100), 25.0 (5/20*100), 75.0 (15/20*100), 85.0 (17/20*100) , 100.0 (20/20*100), respectively.

Reference:

[1] Inna Dubchak, Ilya Muchink, Stephen R.Holbrook and Sung-Hou Kim. Prediction of protein folding class using global description of amino acid sequence. Proc.Natl.Acad.Sci.USA, 1995, 92, 8700-8704.

[2] Inna Dubchak, Ilya Muchink, Christopher Mayor, Igor Dralyuk and Sung-Hou Kim. Recognition of a Protein Fold in the Context of the SCOP classification. Proteins: Structure, Function and Genetics, 1999, 35, 401-407.

1.5 Feature Group 5 [G5]: Quasi-sequence-order descriptors

The quasi-sequence-order descriptors are proposed by K.C.Chou, et.al. They are derived from the distance matrix between the 20 amino acids.

1.5.1 Sequence-order-coupling numbers

The dth-rank sequence-order-coupling number is defined as:

$$\tau_d = \sum_{i=1}^{N-d} \left(d_{i,i+d}\right)^2 \qquad d=1, 2 \ldots \text{maxlag} \tag{13}$$

Where $d_{i,i+d}$ is the distance between the two amino acids at position i and i+d.

Note: Maxlag is the maximum lag and the length of the protein must be not less than maxlag.

1.5.2 Quasi-sequence-order (QSO) descriptors

For each amino acid type, a quasi-sequence-order descriptor can be defined as:

$$Xr = \frac{f_r}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\max lag} \tau_d} \qquad r=1, 2, 3 \ldots 20 \tag{14}$$

Where f_r is the normalized occurrence for amino acid type i and w is a weight factor (w=0.1).

These are the first 20 quasi-sequence-order descriptors. The other 30 quasi-sequence-order are defined as:

$$Xd = \frac{w\tau_{d-20}}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\max lag} \tau_d} \qquad d=21, 22, 23 \ldots 20+\text{maxlag} \tag{15}$$

In addition to Schneider-Wrede physicochemical distance matrix used by Chou et al, another chemical distance matrix by Grantham is also used here.

Reference:

[1] Kuo-Chen Chou. Prediction of Protein Subcellar Locations by Incorporating Quasi-Sequence-Order Effect. Biochemical and Biophysical Research Communications 2000, 278, 477-483.

[2] Kuo C.C. and Yu D.C., Prediction of Protein subcellular locations by GO-FunD-PseAA predictor, Biochemical and Biophysical Research Communications, 2004, 320, 1236-1239.

[3] Gisbert Schneider and Paul wrede. The Rational Design of Amino Acid Sequences by Artifical Neural Networks and Simulated Molecular Evolution: Do Novo Design of an Idealized Leader Cleavge Site. Biophys Journal, 1994, 66, 335-344.

[4] Grantham, R. Amino acid difference formula to help explain protein evolution. Science, 1974, 185, 862-864

1.6 Feature Group 6 [G6]: Pseudo-amino acid composition (PAAC)

This groups of descriptors are proposed by Kuo-chen Chou [1]. PAAC descriptors (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/type1.htm) are also called the type 1 pseudo-amino acid composition. Let $H_1^0(i)$, $H_2^0(i)$, $M^0(i)$ (i=1,2,…,20) be the original hydrophobicity values[2], the original hydrophilicity values[3]and the original side chain masses of the 20 natural amino acids, respectively. They are converted to following qualities by a standard conversion:

$$H_1(i) = \frac{H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20} [H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}]^2}{20}}} \tag{16}$$

$$H_2(i) = \frac{H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20} [H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}]^2}{20}}} \tag{17}$$

$$M(i) = \frac{M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20} [M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}]^2}{20}}} \tag{18}$$

Then, a correlation function can be defines as:

$$\Theta(R_i, R_j) = \frac{1}{3}\{[H_1(R_i) - H_1(R_j)]^2 + [H2(R_i) - H2(R_j)]^2 + [M(R_i) - M(R_j)]^2 \tag{19a}$$

This correlation function is actually an averaged value for the three amino acid properties: hydrophobicity value, hydrophilicity value and side chain mass. Therefore we can extend this definition of correlation function for one amino acid property or for a set of n amino acid properties. For one amino acid property, the correlation can be defined as:

$$\Theta(R_i, R_j) = [H(R_i) - H(R_j)]^2 \tag{19b}$$

Where H (Ri) is the amino acid property of amino acid Ri after standardization.

For a set of n amino acid properties, it can be defined as:

$$\Theta(R_i, R_j) = \frac{1}{n}\sum_{k}^{n}[H_k(R_i) - H_k(R_j)]^2 \tag{19c}$$

Where H_k (Ri) is the kth property in the amino acid property set for amino acid Ri.

A set of descriptors called sequence order-correlated factors are defined as:

$$\begin{aligned}
\theta_1 &= \frac{1}{N-1}\sum_{i=1}^{N-1}\Theta(R_i, Ri+1) \\
\theta_2 &= \frac{1}{N-2}\sum_{i=1}^{N-2}\Theta(R_i, Ri+2) \\
\theta_3 &= \frac{1}{N-3}\sum_{i=1}^{N-3}\Theta(R_i, Ri+3) \\
\theta_\lambda &= \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda}\Theta(R_i, Ri+\lambda), \quad (\lambda < \mathrm{N})
\end{aligned} \tag{20}$$

λ (<L) is a parameter to be chosen. Let f_i is the normalized occurrence frequency of the 20 amino acids

in the protein sequence, a set of 20+λ descriptors called the pseudo-amino acid composition for a protein sequence can be defines as:

$$Xu = \frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda} \theta_j} \quad (1<u<20)$$

(21)

$$Xu = \frac{wf_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda} \theta_j} \quad (20+1 \le u \le 20+\lambda) \tag{22}$$

Where w is the weighting factor for the sequence-order effect and is set as w=0.05 in PROFEAT as suggested by Chou KC [1].

Note: the original hydrophobicity values for amino acids in Profeat are different from the values by Chou KC [1]. In this updated version, the default values of amino acid properties are the values of Chou KC. However, in the work of Chou KC [1-4], the definition for "normalized occurrence frequency" is not given and in this work we define it as the occurrence frequency of amino acid in the sequence normalized to 100% and hence our calculated values are not the same as values by them. [4]

Reference:

[1] Kuo-Chen Chou. Prediction of Protein Cellular Attributes Using Pseudo-Amino Acid Composition. PROTEINS: Structure, Function, and Genetics, 2001, 43:246–255.

[2] Jiri Damborsky. Quantitative structure–function and structure–stability relationships of purposely modified proteins. Protein Engeering, 1998, 11, 21-30

[3] Hopp-Woods. Prediction of protein antigenic determinants from amino acid sequences. Proc. Natl. Acad. Sci. 1981, 78, 3824-3828.

[4] http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/

1.7 Feature Group 7 [G7]: Amphiphilic pseudo-amino acid composition (APAAC)

APAAC (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/type2.htm) are also called type 2 pseudo-amino acid composition. The definitions of these qualities are similar to the above PAAC descriptors. From $H_1(i)$ and $H_2(i)$ defined in eq. 16 and eq. 17, the hydrophobicity and hydrophilicity correlation functions are defined respectively as:

$$H^1_{i,j} = H_1(i)H_1(j)$$

$$H^2_{i,j} = H_2(i)H_2(j) \tag{23}$$

From these qualities, sequence order factors can be defines as:

$$\tau_1 = \frac{1}{N-1}\sum_{i=1}^{N-1} H^1_{i,i+1}$$

$$\tau_2 = \frac{1}{N-1}\sum_{i=1}^{N-2} H^2_{i,i+1}$$

$$\tau_3 = \frac{1}{N-2}\sum_{i=1}^{N-2} H^1_{i,i+2} \qquad (24)$$

$$\tau_4 = \frac{1}{N-2}\sum_{i=1}^{N-2} H^2_{i,i+2}$$

$$\tau_{2\lambda-1} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H^1_{i,i+\lambda}$$

$$\tau_{2\lambda} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H^2_{i,i+\lambda} \qquad (\lambda<N)$$

Then a set of descriptors called “Amphiphilic pseudo amino acid composition” (APAAC) are defined as:

$$p_u = \frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda}\tau_j} \qquad (1< u <20) \qquad (25)$$

$$p_u = \frac{w\tau_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda}\tau_j} \qquad (20+1\le u \le 20+2\lambda) \qquad (26)$$

Where w is the weigh factor and is taken as w=0.5 in PROFEAT as in the work of Chou KC.

Reference:

[1] Kuo-Chen Chou. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics, 2005, 21, 10-19.

1.8 Feature Group 8 [G8]: Topological descriptors at atomic level

Topological descriptors for a molecule can be calculated from the 2D structure. In the updated version of PROFEAT, the 2D structure for a peptide is automatically generated by the program from the peptide sequence. Table 5 gives a list of the topological descriptors that can be calculated by the updated version of PROFEAT.

Table 5 Topological descriptors by PROFEAT

No.	Name
1	Number of Atoms
2	Number of Heavy atoms
3	Number of H atoms
4	Number of B atoms
5	Number of C atoms
6	Number of N atoms
7	Number of O atoms
8	Number of F atoms
9	Number of P atoms
10	Number of S atoms
11	Number of Cl atoms
12	Number of Br atoms
13	Number of I atoms
14	Number of Bonds
15	Number of non-H Bonds
16	Number of rings
17	Molecular weight
18	Average molecular weight
19	Number of H-bond donnor
20	Number of H-bond acceptor
21	Number of 3-member rings
22	Number of 4-member rings
23	Number of 7-member rings
24	Number of 5-member non-aromatic rings
25	Number of 6-member non-aromatic rings
26	Number of 5-member aromatic rings
27	Number of 6-member aromatic rings
28	Number of heterocyclic rings
29	Number of N heterocyclic rings
30	Number of O heterocyclic rings
31	Number of S heterocyclic rings
32	Fingerprint for primary carbocation
33	Fingerprint for secondary carbocation
34	Fingerprint for tertiary carbocation
35	Fingerprint for organohalide
36	Fingerprint for amonium ion
37	fingerprint for primary amonium
38	fingerprint for secondary amonium
39	fingerprint for tertiary amonium
40	fingerprint for nitro
41	fingerprint for nitrile
42	fingerprint for diazo
43	fingerprint for phenol (Ph-OH)
44	fingerprint for primary alcohol
45	fingerprint for second alcohol
46	fingerprint for tertiary alcohol
47	fingerprint for Ph-O-Ph
48	fingerprint for ether(R-O-R)
49	fingerprint for aldehyde(R-CHO)
50	fingerprint for ketone(R-CO-R)
51	fingerprint for carboxylic acid(R-COOH)
52	fingerprint for carboxylate ion (R-COO(-))
53	fingerprint for acyl cation (R-CO(+))
54	fingerprint for ester (R-COOR)
55	fingerprint for Acid anhydride (R-CO-O-COR)
56	fingerprint for Alkoxide ion (R-O(-))
57	fingerprint for peroxide (R-O-O-R)
58	Fingerprint for epoxide (c-O-c ring)
59	Fingerprint for diol (C(OH)-C(OH)-)
60	Fingerprint for organosilicona
61	Fingerprint for organoarsenical
62	Fingerprint for thiol(R-SH)

63	Fingerprint for thiophenol (Ph-SH)
64	Fingerprint t for R-S-R
65	Fingerprint for Ph-S-Ph
66	Fingerprint for sulfonic acid
67	Fingerprint for thioketone (R-C=S)
68	Fingerprint t for phosphonic acid
69	Fingerprint for phosphinic acid
70	Fingerprint for organophophosphate ester
71	Fingerprint for carboxylic thioester
72	Fingerprint for sulfate ester
73	Fingerprint for thiophosphate ester
74	Fingerprint for amide
75	Fingerprint for alpha-amino acid
76	Fingerprint for hydroxynitrile
77	Fingerprint for oxime
78	Fingerprint for nitrate ester
79	Fingerprint for acid halide (RCOX)
80	Number of rotable bonds
81	Schultz molecular topological index
82	Gutman molecular topological index
83	Topological charge index G1
84	Topological charge index G2
85	Topological charge index G3
86	Topological charge index G4
87	Topological charge index G5
88	Mean topological charge index J1
89	Mean topological charge index J2
90	Mean topological charge index J3
91	Mean topological charge index J4
92	Mean topological charge index J5
93	Global topological charge index J
94	Wiener index
95	Mean Wiener index
96	Harary index
97	Gravitational topological index
98	Molecular path count of length 1
99	Molecular path count of length 2
100	Molecular path count of length 3
101	Molecular path count of length 4
102	Molecular path count of length 5
103	Molecular path count of length 6
104	Total path count
105	Xu index
106	Modified Xu Index
107	Balaban Index J
108	Platt Number
109	First Zagreb Index (M1)
110	Second Zagreb Index (M2)
111	First Modified Zagreb Index
112	Second Modified Zagreb Index
113	Quadratic index (Q)
114	0th edge connectivity index
115	Edge connectivity index
116	Extened edge connectivity inndex
117	0th Kier-Hall connectivity index
118	1th Kier-Hall connectivity index
119	Mean Randic Connectivity index
120	2th Kier-Hall connectivity index
121	Simple topological index by Narumi
122	Harmonic topological index by Narumi
123	Geometric topological index by Narumi
124	Arithmetic topological index by Narumi
125	0th valence connectivity index
126	1th valence connectivity index
127	2th valence connectivity index

128	0th order delta chi index
129	1th order delta chi index
130	2th order delta chi index
131	Pogliani index
132	0th Solvation connectivity index
133	1th Solvation connectivity index
134	2th Solvation connectivity index
135	1th order Kier shape index
136	2th order Kier shape index
137	3th order Kier shape index
138	1th order Kappa alpha shape index
139	2th order Kappa alpha shape index
140	3th order Kappa alpha shape index
141	ier Molecular Flexibility Index
142	Topological radius
143	Topological diameter
144	Graph-theoretical shape coefficient
145	Eccentricity
146	Average atom eccentricity
147	Mean eccentricity deviation
148	Average distance degree
149	Mean distance degree deviation
150	Unipolarity
151	Rouvary index
152	Centralization
153	Variation
154	Dispersion
155	Log of PRS INDEX
156	RDSQ ondex
157	RDCHI index
158	Optimized 1th connectivity index
159	Logp from connectivity
160-219	BCUT descriptors
220-285	Moreau-Broto Autocorrelation descriptors
286-345	Moran Autocorrelation descriptors
346-405	Geary Autocorrelation descriptors
406	Topological polar surface area (TPSA)

Reference:

[1] Roberto T. and Viviana C., Handbook of Molecular Descriptors, Wiley-VCH, 2000.

1.9 Feature Group 9 [G9]: Total amino acid properties (TAAP)

The "total amino acid property (TAAP)" descriptor for a property i is defined here as:

$$P_{tot(i)} = \sum_{j=1}^{N} p_j^i \tag{27}$$

Where p_j^i is the property i of amino acid R_j and N is the length of the sequence.

Reference:

[1] M.Michael G., Makiko S. Influence of amino acid properties for discriminating outer membrane proteins at better accuracy. Biochimica of Biophysica Acta, 2006, 1493-1497.

Table 6 List of the descriptors for proteins or peptides

No.	Descriptors type	Number of descriptors	Symbol
1	Amino acid composition	20	f(i)
2	Dipeptide composition	400	fr(i,j)
3	Moreau-Broto autocorrelation descriptor	Dependent on number of properties	ATS(d)
4	Moran autocorrelation descriptor	Dependent on number of properties	I(d)
5	Geary autocorrelation descriptor	Dependent on number of properties	C(d)
6	Composition Transition Distribution	72	C
7	Transition	72	T
8	Distribution	360	D
9	Sequence-order-coupling number	90	τ_d
10	Quase-sequence-order descriptors	150	Xd
11	Pseudo-amino acid composition	Dependent on number of property sets and lamda	Xu
12	Amphiphilic pseudo-amino acid composition	Dependent on lamda	Pu
13	Topological descriptors	405	X_{top}
14	Total amino acid prpperties	Dependent on number of properties	P_{tot}

2. Ligand (Small Molecule) Descriptors

These descriptors are topological descriptors for ligands or small molecules calculated from the 2D structure in sdf or mol format. The names of these descriptors for a ligand are the same as for a proteins or a peptide listed in Table 5.

3. Protein-Protein Interaction Pair Descriptors

Let $V_a=\{V_a(i), i=1,2…n\}$ and $V_b=\{Vb(i), i=1,2…n\}$ are the two descriptor vectors for interaction protein A and protein B, respectively, then there are 3 methods to construct the descriptor vector V for A and B:

(1) Two vectors Vab and Vba with dimension of 2n are constructed: $V_{ab}=(V_a, V_b)$ for interaction between protein A and protein B and $V_{ba}=(V_b, V_a)$ for interaction between protein B and protein A.

(2) One vector V with dimension of 2n is constructed:

$V=\{V_a(i)+V_b(i), V_a(i) \times V_b(i), i=1,2 … n\}$.

(3) One vector V with dimension of n^2 is constructed by the tensor product:

$V=\{V(k) = V_a(i)\times V_b(j), i=1, 2 … n; j=1, 2 … n; k=(i-1)\times n+j\}$.

4. Protein-Ligand Interaction Pair Descriptors

There are two methods for construction of descriptor vector **V** for protein-ligand interaction from protein descriptor vector $\mathbf{V_p}$ ($V_p(i)$, $i=1,n_p$) and ligand descriptor vector $\mathbf{V_l}$ ($V_l(i)$, $i=1,n_l$)**:**

(1) One vector V with dimension of np+nl are constructued: V=($\mathbf{V_p}$, $\mathbf{V_l}$) for interaction between protein P and Ligand L.

(2) One vector V with dimension of $n_p \times n_l$ is constructed by the tensor product:

$\mathbf{V}=\{v(k)=V_p(i) \times V_l(j), i=1,2…n_p, j=1,2…n_l, k=(i-1) \times n_p+j\}$.

5. Biological Network Descriptors

In the following table, all the descriptors are grouped into different feature categories according to their algorithm definitions, and each column lists the computed descriptors for each input network type. Some descriptors can be defined by either un-weighted connection information or weighted information. Therefore, some notations are given: "○" represents the features calculated based on un-weighted network structure, "★" represents the features calculated based on edge weight, "●" represents the features calculated based on node weight, and "➚" represents the features calculated based on directed information.

ID	Feature Category	Network Descriptor Name	Network Type			
			Un-Directed			Directed
			Un-Weighted	Edge Weighted	Node Weighted	Un-Weighted
Node-Level Descriptors [Local]						
1	Connectivity Profiles to the Immediate Neighbors	Degree	○	○	○	
2		Scaled Connectivity	○	○	○	
3		Number of Selfloops	○	○	○	➚
4		Number of Triangles	○	○	○	➚
5		Z Score	○	○	○	
6	Connectivity Profiles to the Next Immediate Neighbors	Clustering Coefficient	○	○	○	
7		Neighborhood Connectivity	○	○	○	
8		Topological Coefficient	○	○	○	
9		Interconnectivity	○	○	○	
10		Bridging Coefficient	○	○	○	
11	Distance Relationships to All Other Nodes	Average Shortest Path Length	○	○ ★	○	
12		Distance Sum	○	○ ★	○	
13		Eccentricity	○	○ ★	○	
14		Eccentric	○	○ ★	○	
15		Deviation	○	○ ★	○	
16		Distance Deviation	○	○ ★	○	
17		Radiality	○	○ ★	○	
18	Centrality based on Degree or Distance to all Other Nodes	Degree Centrality	○	○	○	
19		Closeness Centrality (avg)	○	○ ★	○	
20		Closeness Centrality (sum)	○	○ ★	○	
21		Eccentricity Centrality	○	○ ★	○	
22		Harmonic Closeness Centrality	○	○ ★	○	
23		Residual Closeness Centrality	○	○ ★	○	
24	Centrality based on Shortest Paths Passing thru the Studied Node	Stress Centrality	○	○ ★	○	
25		Betweenness Centrality	○	○ ★	○	
26		Normalized Betweenness	○	○ ★	○	
27		Bridging Centrality	○	○ ★	○	
28	Centrality based on Connectivity and Neighbors' Centrality	Page Rank Centrality	○	○	○	
29		Eigenvector Centrality	○	○	○	
30	Edge-Weighted Descriptor	Strength		★		
31		Assortativity		★		
32		Disparity		★		

33		Geometric Mean of Triangles		★		
34		Barrat's Local Clustering Coefficient		★		
35		Onnela's Local Clustering Coefficient		★		
36		Zhang's Local Clustering Coefficient		★		
37		Holme's Local Clustering Coefficient		★		
38	Node-Weighted Descriptor	Node Weight			●	
39		Node Weighted Cross Degree			●	
40		Node Weighted Local Clustering Coefficient			●	
41	Directed Descriptor	In-Degree				↗
42		Out-Degree				↗
43		Directed Local Clustering Coefficient				↗
44		Neighbourhood Connectivity (only in)				↗
45		Neighbourhood Connectivity (only out)				↗
46		Neighbourhood Connectivity (in & out)				↗
47		Average Directed Neighbour Degree				↗
Node-Level Descriptors [Global]						
1	Basic Global Connectivity Profiles	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	Network Measure Based on all Shortest Paths	Total Distance	○	○ ★		
15		Network Diameter	○	○ ★		
16		Network Radius	○	○ ★		
17		Shape Coefficient	○	○ ★		
18		Characterisitc Path Length	○	○ ★		
19		Network Eccentricity	○	○ ★		
20		Average Eccentricity	○	○ ★		
21		Network Eccentric	○	○ ★		
22		Eccentric Connectivity	○	○ ★		
23		Unipolarity	○	○ ★		
24		Integration	○	○ ★		
25		Variation	○	○ ★		
26		Average Distance	○	○ ★		
27		Mean Distance Deviation	○	○ ★		
28		Centralization	○	○ ★		
29		Global Efficiency	○	○ ★		
30	Topological Index Based on Connectivity	Edge Complexity Index	○	○		
31		Randic Connectivity Index	○	○		
32		Atom-Bond Connectivity Index	○	○		

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

81		Bonchev Information Index 1	○	○		
82		Bonchev Information Index 2	○	○		
83		Bonchev Information Index 3	○	○		
84		Balaban-like Information Index 1	○	○		
85		Balaban-like Information Index 2	○	○		
86	Eigenvalue-Based Connectivity Descriptors	Graph Energy	○	○		
87		Laplacian Energy	○	○		
88		Spectral Radius	○	○		
89		Estrada Index	○	○		
90		Laplacian Estrada Index	○	○		
91		Quasi-Weiner Index	○	○		
92		Mohar Index 1	○	○		
93		Mohar Index 2	○	○		
94		Graph Index Complexity	○	○		
95		Adjacency Matrix HM	○	○		
96		Adjacency Matrix SM	○	○		
97		Adjacency Matrix ISM	○	○		
98		Adjacency Matrix PM	○	○		
99		Adjacency Matrix IPM	○	○		
100		Laplacian Matrix HM	○	○		
101		Laplacian Matrix SM	○	○		
102		Laplacian Matrix ISM	○	○		
103		Laplacian Matrix PM	○	○		
104		Laplacian Matrix IPM	○	○		
105		Distance Matrix HM	○	○		
106		Distance Matrix SM	○	○		
107		Distance Matrix ISM	○	○		
108		Distance Matrix PM	○	○		
109		Distance Matrix IPM	○	○		
110		Distance Path Matrix HM	○	○		
111		Distance Path Matrix SM	○	○		
112		Distance Path Matrix ISM	○	○		
113		Distance Path Matrix PM	○	○		
114		Distance Path Matrix IPM	○	○		
115		Augmented Vertex Degree Matrix HM	○	○		
116		Augmented Vertex Degree Matrix SM	○	○		
117		Augmented Vertex Degree Matrix ISM	○	○		
118		Augmented Vertex Degree Matrix PM	○	○		
119		Augmented Vertex Degree Matrix IPM	○	○		
120		Extended Adjacency Matrix HM	○	○		
121		Extended Adjacency Matrix SM	○	○		
122		Extended Adjacency Matrix ISM	○	○		
123		Extended Adjacency Matrix PM	○	○		
124		Extended Adjacency Matrix IPM	○	○		
125		Vertex Connectivity Matrix HM	○	○		
126		Vertex Connectivity Matrix SM	○	○		
127		Vertex Connectivity Matrix ISM	○	○		
128		Vertex Connectivity Matrix PM	○	○		

129		Vertex Connectivity Matrix IPM	○	○		
130		Random Walk Markov Matrix HM	○	○		
131		Random Walk Markov Matrix SM	○	○		
132		Random Walk Markov Matrix ISM	○	○		
133		Random Walk Markov Matrix PM	○	○		
134		Random Walk Markov Matrix IPM	○	○		
135		Weighted Struct. Func. Matrix IM1 HM	○	○		
136		Weighted Struct. Func. Matrix IM1 SM	○	○		
137		Weighted Struct. Func. Matrix IM1 ISM	○	○		
138		Weighted Struct. Func. Matrix IM1 PM	○	○		
139		Weighted Struct. Func. Matrix IM1 IPM	○	○		
140		Weighted Struct. Func. Matrix IM2 HM	○	○		
141		Weighted Struct. Func. Matrix IM2 SM	○	○		
142		Weighted Struct. Func. Matrix IM2 ISM	○	○		
143		Weighted Struct. Func. Matrix IM2 PM	○	○		
144		Weighted Struct. Func. Matrix IM2 IPM	○	○		
145	Edge-Weighted Descriptors	Weighted Transitivity		★		
146		Barrat's Global Clustering Coefficient		★		
147		Onnela's Global Clustering Coefficient		★		
148		Zhang's Global Clustering Coefficient		★		
149		Holme's Global Clustering Coefficient		★		
150	Node-Weighted Descriptor	Total Node Weight			●	
151		Node Weighted Global Clustering Coefficient			●	
152	Directed Descriptor	Average In-Degree				➚
153		Maximum In-Degree				➚
154		Minimum In-Degree				➚
155		Average Out-Degree				➚
156		Maximum Out-Degree				➚
157		Minimum Out-Degree				➚
158		Directed Global Clustering Coefficient				➚

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 node *j*.
   2.2. Normalized edge weight matrix "*NorEW*", by the following definition. Here, the constant factor 0.99 in the denominator is to slightly enlarge the domain from minimum value to maximum value, such that ensure 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*", as defined below. Again, the constant factor 0.99 in the denominator is to slightly enlarge the domain from minimum value to maximum value, such that 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 the table given previously. As some descriptors can be derived from either un-weighted adjacency matrix or weight 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.

5.1 Feature Group 10 [G10]: Node-Level Descriptors

Feature Category: Connectivity Profiles to the Immediate Neighbours

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[1]

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

5. Z Score[2,3]

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.

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

Feature Category: Connectivity Profiles to the Next Immediate Neighbours

6. Clustering Coefficient[4,5]

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*. The nodes with less than two neighbours have its value assumed to be 0.

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

7. Neighborhood Connectivity[6]

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[7]

The topological coefficient of a node *i* is defined as below, where *j* represents all the nodes sharing at least one neighbour with *i*, and *J(i, j)* is the number of shared neighbours between node *i* and *j*.

If there is a direct edge between *i* and *j*, plus an additional one to *J(i, j)*. It is a measure to estimate the tendency of the nodes to have shared neighbours in the network.

$$topology_i = avg\left\{\frac{J(i,j)}{deg_i}\right\}$$

9. Interconnectivity[8,9,10]

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[11]

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

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

Feature Category: Distance Relationships to All Other Nodes

11. Average Shortest Path Length[12]

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.

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

12. Distance Sum[13]

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

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

13. Eccentricity[13]

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

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

14. Eccentric[13]

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

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

15. Deviation[13]

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.

$$deviation_i = distSum_i - unipolarity_G$$

16. Distance Deviation[13]

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

$$distDev_i = |distSum_i - distAvg_G|$$

17. Radiality[14]

Radiality is computed by subtracting the average shortest path length of node *i* from the network 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.

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

Feature Category: Centrality Based on Degree or Distance to All Other Nodes

18. Degree Centrality[15]

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

19. Closeness Centrality (avg)[14,16,17]

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[18]

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[19]

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

Feature Category: Centrality Based on Shortest Paths Passing Through the Studied Node

24. Stress Centrality[14,20]

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

Here, *s* and *t* are the nodes different from *i* in the network, and σ_{st} *(i)* is the number of shortest paths from *s* to *t* that passing through *i*.

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

25. Betweenness Centrality[14,21]

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, t,* σ_{st} *(v)* are defined as same as the previous stress centrality, 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[11]

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.

$$centralityBridge_i = bridge_i \cdot centralityBtw_i$$

Feature Category: Centrality Based on Connectivity and Neighbors' Centrality

28. PageRank Centrality[22,23,24,25,26,27]

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 (see next), by initializing the PageRank centralities to an equal probability value *1/N* for all nodes. The equation below 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 constant damping factor *d* is generally assumed to 0.85.

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

29. Eigenvector Centrality[28,29]

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.

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

Feature Category: Edge-Weighted

30. Strength[30]

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

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

31. Assortativity[30,31]

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

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

32. Disparity[32]

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

33. Geometric Mean of Triangles[1]

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

34. Barrat's Local Clustering Coefficients[33]

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

35. Onnela's Local Clustering Coefficients[33,34]

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

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

36. Zhang's Local Clustering Coefficients[33,35]

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

37. Holme's Local Clustering Coefficients[33,36]

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

Feature Category: Node-Weighted

38. Node Weight

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

39. Node Weighted Cross Degree[37]

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.

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

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

40. Node Weighted Local Clustering Coefficient[37]

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.

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

Feature Category: Directed

41. In-Degree[1,38]

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.

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

42. Out-Degree[1,38]

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

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

43. Directed Local Clustering Coefficient[38]

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

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

44. Neighbourhood Connectivity (only in)[38]

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

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

45. Neighbourhood Connectivity (only out)[38]

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

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

46. Neighbourhood Connectivity (in & out)[38]

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

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

47. Average Directed Neighbour Degree[1]

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

5.2 Feature Group 11 [G11]: Network-Level Descriptors

Feature Category: Basic Global Connectivity Profiles

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

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

7. Total Adjacency[39]

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[39]

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[4,5]

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[1]

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

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

11. Heterogeneity[40]

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[40]

Degree centralization is to distinguish 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[41]

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

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

Feature Category: Network Measure Based on All Shortest Paths

14. Total Distance[39]

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

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

15. Network Diameter

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

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

16. Network Radius

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

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

17. Shape Coefficient [42]

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

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

18. Characterisitc Path Length

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

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

19. Network Eccentricity[13]

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

20. Average Eccentricity[13]

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

21. Network Eccentric[13]

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

22. Eccentric Connectivity[43]

This index 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.

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

23. Unipolarity[13]

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

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

24. Integration[13]

Network integration is the sum of all nodes' distance sum, where each shorest path is counted once.

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

25. Variation[13]

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

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

26. Average Distance[13]

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

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

27. Mean Distance Deviation[13]

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

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

28. Centralization[13]

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

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

29. Global Efficiency[44]

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.

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

Feature Category: Topological Index Based on Connectivity

30. Edge Complexity Index[39]

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

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

31. Randic Connectivity Index[45]

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

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

32. Atom-Bond Connectivity Index[46]

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.

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

33. Zagreb Index 1[47,48,49,50]

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

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

34. Zagreb Index 2

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

35. Modified Zagreb Index

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

36. Augmented Zagreb Index

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

37. Variable Zagreb Index

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

38. Narumi-Katayama Index[51]

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.

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

39. Narumi-Katayama Index (log)

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

40. Narumi Geometric Index[52]

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

41. Narumi Harmonic Index[52]

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

42. Alpha Index[3]

The 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.

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

43. Beta Index[3]

It measures the network 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.

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

44. Pi Index[3]

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.

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

45. Eta Index[3]

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

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

46. Hierarchy[3]

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.

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

47. Robustness[53]

Robustness is to measure the stability of a network under node-removal attacks. Under the persistent attack, the size of the largest fragmented component *S* and the number of nodes removed *k* are used to define the robustness. An ideally robust network has its largest component *S* decrease linearly, but a fragile network collapses much faster, and the drop of value *S* will indicate this collapse.

$$robustness_G = \frac{600 \cdot \sum_{k=1}^{N} k \cdot S_k}{N(N+1)(N-1)}$$

48. Medium Articulation[54,55,56]

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:

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

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

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

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

Mutual information MA_I is defined as:

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

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

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

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

Feature Category: Topological Index Based on Shortest Path Distances

49. Complexity Index A[39]

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

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

50. Complexity Index B[39]

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

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

51. Wiener Index[57]

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

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

52. Hyper-Wiener Index[58]

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

53. Harary Index 1[59]

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

54. Harary Index 2[59]

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

55. Compactness[60]

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

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

56. Superpendentic Index[61]

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

57. Hyper-Distance-Path Index[62,63]

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

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

58. BalabanJ Index[64]

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.

59. BalabanJ-Like Index 1[65]

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

60. BalabanJ-Like Index 2[65]

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

61. BalabanJ-Like Index 3[65]

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

62. Geometric Arithmetic Index 1[48,66]

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

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

63. Geometric Arithmetic Index 2[48,66]

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.

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

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

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

In the definition of GA index 2, *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.

64. Geometric Arithmetic Index 3[48,66]

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

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

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

In the definition of GA index 3, *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.

65. Szeged Index[67]

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

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

66. Product of Row Sums[68]

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

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

67. Product of Row Sums (log)

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

68. Schultz Topological Index[69]

By using adjacency matrix *A*, shorest 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.

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

69. Gutman Topological Index[70]

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

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

70. Efficiency Complexity[54,55,56]

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

71. Information Content (Degree Equality)[71]

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.

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

72. Information Content (Edge Equality)[72]

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.

73. Information Content (Edge Magnitude)[72]

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.

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

74. Information Content (Distance Degree)[71]

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

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

75. Information Content (Distance Degree Equality)[71]

The probability distribution regarding on the nodes' distance degree value gives the definition of the mean information content on distance degree equality.

In the equation 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.

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

76. Radial Centric Information Index[71]

It is a descriptor 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.

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

77. Distance Degree Compactness[73]

This measure is defined based on the distribution of nodes' locations from the center of a network, where the center is determined by the closeness centrality score in this case.

Here, Q_k is the sum of distance degree of all nodes located at the same topological distance k from the center.

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

78. Distance Degree Centric Index[74]

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

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

79. Graph Distance Complexity[75]

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

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

80. Information Layer Index[76]

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

In the equation, 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.

81. Bochev Information Index 1[77]

Bochev indices applies the probability distribution of the shortest path lengths to the Shannon's entropy formula, and it has three different variants.

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

Where $diameter_G$ 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}.

82. Bochev Information Index 2[77]

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

83. Bochev Information Index 3[77]

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

84. Balaban-like Information Index 1[78,79]

Previously, the BalabanJ indices were defined by the distance degree of each node. Here, Balaban-like information index 1 & 2 are defined based on the distribution of distance degree in the network.

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

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

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

Where g_k is the number of nodes that are at distance k from node i, and μ is namely the cyclomatic number.

85. Balaban-like Information Index 2[78,79]

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

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

Feature Category: Eigenvalue-Based Complexity

86. Graph Energy[80]

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

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

87. Laplacian Energy[80]

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

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

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

88. Spectral Radius[81]

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

89. Estrada Index[82]

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

90. Laplacian Estrada Index[83]

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

91. Quasi-Wiener Index[63,84]

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

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

92. Mohar Index 1[63,84]

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

93. Mohar Index 2[63,84]

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

94. Graph Index Complexity[54]

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

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

95 - 144. A Set of Eigenvalue-Based Descriptors from Variants of Matrices [85,86]

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. Additionally, $s = 1$ for the graph with odd number of nodes, and $s = 2$ for the graph with even number of nodes.

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

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

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

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

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

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 matrix.

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

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

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

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

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

(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.

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

(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

145. Weighted Transitivity[1]

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

146. Barrat's Global Clustering Coefficients[33]

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

147. Onnela's Global Clustering Coefficients[33,34]

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

148. Zhang's Global Clustering Coefficients[33,35]

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

149. Holme's Global Clustering Coefficients[33,36]

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

Feature Category: Node-Weighted

150. Total Node Weight

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

151. Node Weighted Global Clustering Coefficient[37]

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

Feature Category: Directed

152. Average In-Degree

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

153. Maximum In-Degree

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

154. Minimum In-Degree

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

155. Average Out-Degree

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

156. Maximum Out-Degree

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

157. Minimum Out-Degree

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

158. Directed Global Clustering Coefficient[38]

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

Reference

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

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

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

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

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

6 Maslov, S. & Sneppen, K. Specificity and Stability in Topology of Protein Networks. *Science*, 296, 910-913, (2002).

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

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

9 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).

10 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).

11 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).

12 Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1, 269-271, (1959).

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

14 Brandes, U. A Faster Algorithm for Betweenness Centrality. *J Math Sociol*, 25, (2001).

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

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

17 Newman, M. E. J. A Measure of Betweenness Centrality Based on Random Walks. *Social Networks*, 27, (2003).

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

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

20 Shimbel, A. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15, 501-507, (1953).

21 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).

22 Langville, A. N. & Meyer, C. D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. (Princeton University Press, 2012).

23 Michael, W. B. & Murray, B. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Vol. 8 (Society for Industrial and Applied Mathematics, 1999).

24 Brin, S. & Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *7th International World-Wide Web Conference*, (1998).

25 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).

26 Gleich, D. F. PageRank beyond the Web. *SIAM Rev*, 57, 321-363, (2014).

27 Ivan, G. & Grolmusz, V. When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics*, 27, 405-407, (2011).

28 Straffin, P. D. Linear Algebra in Geography: Eigenvectors of Networks. *Mathematics Magazine*, 53, 269-276, (1980).

29 Newman, M. E. J. *Mathematics of Networks*. 2nd edn, (Palgrave Macmillan, 2008).

30 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).

31 Leung, C. C. & Chau, H. F. Weighted Assortative And Disassortative Networks Model. *Physica A: Statistical Mechanics and its Applications*, 378, 591-602, (2008).

32 Barthélemy, M. Architecutre of Complex Weighted Networks. *Talk Series on Networks and Complex Systems*, (2005).

33 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).

34 Onnela, J. P., Saramäki, J., Kertész, J. & Kaski, K. Intensity and Coherence of Motifs in Weighted Complex Networks. *Physical Review E*, 71, (2005).

35 Zhang, B. & Horvath, S. A General Framework for Weighted Gene Co-Expression Network Analysis. *Stat Appl Genet Mol Biol*, 4, Article17, (2005).

36 Holme, P., Park, S. M., & 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).

37 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).

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

39 Bonchev, D. *Complexity in Chemistry, Biology, and Ecology*. (Springer US, 2007).

40 Dong, J. & Horvath, S. Understanding network concepts in modules. *BMC Syst Biol*, 1, 24, (2007).

41 Mangioni, G. *Complex Networks: Results of the 1st International Workshop on Complex Networks*. 1st edn, Vol. 207 (Springer-Verlag Berlin Heidelberg, 2009).

42 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).

43 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).

44 Latora, V, M. Efficient Behavior of Small-World Networks. *Physical Review Letters*, 87, (2001).

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

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

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

48 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).

49 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).

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

51 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).

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

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

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

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

56 Dehmer, M., Grabner, M. & Varmuza, K. Information Indices with High Discriminative Power for Graphs. *PLoS One*, 7, (2012).

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

58 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).

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

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

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

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

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

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

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

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

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

68 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).

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

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

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

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

73 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).

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

75 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).

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

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

78 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).

79 Dehmer, M. & Mowshowitz, A. A History of Graph Entropy Measures. *Information Sciences*, 181, 57-78, (2011).

80 Gutman, I. & Zhou, B. Laplacian Energy of a Graph. *Linear Algebra and its Applications*, 414, 29-37, (2006).

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

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

83 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).

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

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

86 Dehmer, M. Uniquely Discriminating Molecular Structures Using Novel Eigenvalue-Based Descriptors. *MATCH Commun. Math. Comput. Chem.*, 67, 147-172, (2012).