



MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES



DEVELOPMENT OF A WEB—BASED TOOL FOR NETWORK ANALYSIS OF PROTEIN STRUCTURES

RASİM MURAT AYDINKAL

MASTER THESIS

Department of Bioengineering

Thesis Supervisor

Assoc. Prof. Dr. Pemra ÖZBEK SARICA

ISTANBUL, 2019

**MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE
STUDIES IN PURE AND APPLIED
SCIENCES**

Rasim Murat AYDINKAL, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended his thesis entitled “**Development of a web-based tool for network analysis of protein structures**”, on Jan 28, 2019 and has been found to be satisfactory by the jury members.

Jury Members

Assoc.Prof. Dr. Pemra ÖZBEK SARICA (Advisor)
Marmara University



Assoc.Prof. Dr. Kazım Yalçın ARĞA (Jury Member)
Marmara University



Assoc.Prof. Dr. Elife Zerrin BAĞCI (Jury Member)
Tekirdağ Namık Kemal University



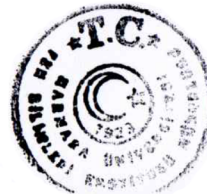
APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Rasim Murat AYDINKAL be granted the degree of Master of Science in Department of Bioengineering, Bioengineering Program on

06.02.2019

(Resolution no: 2019/04-02).


Director of the Institute
Prof. Dr. Bülent EKİCİ





**MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES**



**DEVELOPMENT OF A WEB—BASED TOOL FOR
NETWORK ANALYSIS OF PROTEIN STRUCTURES**

RASİM MURAT AYDINKAL
(524214009)

MASTER THESIS

Department of Bioengineering

Thesis Supervisor

Assoc. Prof. Dr. Pemra ÖZBEK SARICA

ISTANBUL, 2019

ACKNOWLEDGEMENTS

The work presented in this thesis was conducted in the Computational Biology and Bioinformatics Laboratory at the Bioengineering Department of Marmara University. I have learned a lot and really enjoyed while working on this thesis. This thesis becomes a reality with the valuable support and great help of many individuals. I would like to extend my sincere thanks to all of them.

Foremost, I would like to express my gratitude to my thesis supervisor Assoc. Prof. Dr. *Pemra Özbek Sarıca* for her continuous guidance, full support and encouragement.

I am grateful to Dr. *Onur Serçinoğlu* for his patience and for pointing out key points throughout this study. My special thanks to Assoc. Prof. Dr. *Elife Zerrin Bağcı* for pioneering me to improve my thinking abilities. I admire her scientific perspective when we work together. Her guidance helped me during the entire course of research and writing of my academic life. My sincere thanks also goes to Dr. *Yasin Bakış* for mentoring me in all my professional life and for promoting me in Bioinformatics field. He always had a word of encouragement for me. I would like to thank to Mr. *Marco Biasini* from Swiss Institute of Bioinformatics and Biozentrum at University of Basel, for his precious help with WebGL protein viewer despite his super busy schedule.

I extend my gratitude to the executives, founder and board members of Ali Nihat Gökyiğit Foundation, Mr. *Salih Sercan Kanoğlu*, Prof. Dr. *Adil Güner*, Mr. *Ali Nihat Gökyiğit*, Mrs. *Vahide Gigin* and Mr. *Turgut Gökyiğit* for their support during my master studies.

Lastly I am indebted to my wife, *Demet* had taken extra burdens on my behalf to take care of numerous familial tasks for several months while extending her full support and allowing me to focus on this thesis. This study would not have taken place without her love and patience.

This work is dedicated to my family, *Demet* and my 7-month-old superhero *Ahmet Alp*.

January, 2019

Rasim Murat Aydıncal

CONTENTS

	Page
ACKNOWLEDGEMENTS.....	i
CONTENTS.....	ii
ÖZET	v
ABSTRACT	vi
LIST OF SYMBOLS	vii
LIST OF ABBREVIATIONS.....	viii
LIST OF FIGURES.....	x
LIST OF TABLES.....	xiii
CHAPTER 1 INTRODUCTION	14
1.1. Networks of Protein Structures.....	14
1.2. Protein Interaction Networks and Related Definitions	16
1.3. Network Parameters	18
1.4. Visualization of Biological Networks	23
1.5. Computational Tools.....	25
1.5.1. Network visualization tools.....	25
1.5.1.1. Cytoscape.....	26
1.5.1.2. Other tools.....	27
1.5.2. Tools for topological analysis of protein interactions	29
1.5.2.1. NAPS: Network Analysis of Protein Structures	30
1.5.2.2. RING2: Residue Interaction Network Generator.....	31
1.5.2.3. NetworkX	33
1.5.2.4. RINalyzer: Residue Interaction Network Analyzer	33
1.6. Motivation of Developing ProSNEx.....	33
1.7. Aim of the Study	36
CHAPTER 2 MATERIALS AND METHODS	38
2.1. Materials	38
2.2. Methods	38
2.2.1. Data input.....	38
2.2.2. Identifying the network parameters	39
2.2.3. Network construction and visualization.....	39

2.2.4. Centrality analysis	39
2.2.5. Shortest path analysis.....	40
2.2.5.1. Dijkstra's shortest path algorithm.....	40
2.2.5.2. Floyd–Warshall algorithm.....	40
2.2.5.3. Bellman–Ford algorithm.....	40
2.3. Scatter Plot, Contact Map, Distance Map	41
2.4. <i>k</i> -clique Concept	43
2.5. Edge Weight in ProSNEx.....	43
CHAPTER 3 RESULTS AND DISCUSSION.....	44
3.1. Select PDB File.....	44
3.2. Network Construction Settings.....	45
3.3. Main Screen: 3D Structure of the Molecule, Residue Interaction Network and Analysis Tools	46
3.3.1. 3D structure of the molecule	46
3.3.2. Residue interaction network.....	46
3.3.3. Analysis tools	47
3.3.4. About and help panels.....	48
3.3.5. Event logs.....	48
3.3.6. Configurations panel.....	48
3.3.7. View.....	48
3.3.8. Analysis tools panel.....	48
3.3.8.1. Topological parameters	49
3.3.8.2. Preferences.....	51
3.3.8.3. Comparison.....	56
3.3.8.4. 3D structure coloring.....	57
3.3.8.5. Additional features	60
3.3.8.6. Shortest path algorithm.....	60
3.3.8.7. Edge weight	61
3.3.8.8. Display map	62
3.3.8.9. Node centrality analysis.....	64
3.3.8.10. <i>k</i> -clique analysis	68
CHAPTER 4 CONCLUSION	70

REFERENCES.....72
APPENDIX.....82
AUTOBIOGRAPHY 85



ÖZET

PROTEİN YAPILARININ AĞ ANALİZİ İÇİN WEB TABANLI BİR ARAÇ GELİŞTİRİLMESİ

Proteinlerin iç iletişiminin anlaşılması, proteinlerin fonksiyonlarını belirlemek için önemlidir. Her ne kadar sadece birkaç amino asit bağlanma ve sinyal iletimi gibi işlevlerde rol alıyor gibi görünse de, proteinin büyük bir kısmı bu fonksiyonun gerçekleşebilmesi için görev almaktadır. Bu nedenle, proteinin yapısını bir ağ olarak düşünmek ve bu ağın küresel ağ özelliklerini ortaya çıkarmak oldukça yaygın bir düşüncedir.

Protein yapı ağları oluşturmak ve analiz etmek için bazı web servisleri mevcuttur. Ancak ağ analizi için mevcut web uygulamaları, temel geometrik ağ oluşturma ve analizi seçenekleri gibi oldukça kısıtlı işlevler sunmaktadır. Bu çalışmada, yenilikçi ve kapsamlı analiz özellikleri sunan, ProSNEx (Protein Structure Network Explorer) olarak adlandırdığımız web tabanlı çalışan bir araç geliştirilmiştir.

Bu çalışmanın en büyük özgülüğü herhangi bir programın yüklenmesine gerek duyulmadan analizlerin web ortamında yapılabilmesine imkan tanıyan bütünsel bir yaklaşımın hayata geçirilmesi olacaktır. Hem istenen farklı girdilere göre ağ yapısı oluşturulabilecek, hem bu ağın incelenmesi için tüm ağ hesaplarının (düğüm analizleri, derece, yakınlık, arasındalık) yapılmasına olanak sağlayan, bulguların hem 3-boyutlu hem de 2-boyutlu ortamda görselleştirilmesine imkan veren, verilerin sayısal olarak tablo ve grafiklerle gösterimini sağlayıp indirilmesine imkan tanıyan ve bunların üstüne ek olarak UniProt alt yapısını kullanarak Pfam ve doğal varyant verilerine de ulaşmayı ve kullanmayı sağlayan çok amaçlı bir program hazırlanmıştır. Normal mod analizinden çapraz korelasyonları veya gRINN uygulamasının çıktılarından etkileşim enerjisi verilerini düğümler arası bağ ağırlığı olarak belirleme özellikleri ile ağırlıklı ağ (weighted network) oluşturmayı da mümkün kılmaktadır. Bu web aracına <http://www.prosnex-tool.com> adresinden ulaşılabilir.

Ocak, 2019

Rasim Murat Aydıncal

ABSTRACT

DEVELOPMENT OF A WEB-BASED TOOL FOR NETWORK ANALYSIS OF PROTEIN STRUCTURES

To determine the function of proteins, understanding the interaction within proteins is crucial. Although, only a few amino acids are expected to play a role in the function as in binding or signal transmission, a rather large fraction of the protein is participating to its function at the background. Thus, it is proper to consider a protein structure as a connected network and then deduce its related properties that describe the global network features.

We have developed a web-based application called ProSNEx (Protein Structure Network Explorer), which provides a novel and enhanced analysis of protein structures using network formalism.

The highlighted feature of this web tool is that all of the analysis can be carried out on the web environment without installing any other dependencies, as this tool offers a complete solution. It allows to perform node centrality, degree centrality, betweenness centrality on the generated network. In addition to these features, it allows to access and use of Pfam and natural variant annotations from UniProt database. Weighted networks can also be constructed by using either the residue-residue interaction energies in the format returned by gRINN or the dynamical cross correlations from Normal Mode Analysis (NMA) of the protein structure which are automatically retrieved from the PDBj database. As this web-tool will be free and open to all users and there will be no login requirement, there will be numerous achievements to the researchers related to the subject. In particular, this web-tool will highlight the residues which have key role in the signal transduction. It will allow users to focus on the signal transduction pathway, which will offer infinite benefits for allosteric studies. The tool can be accessed at <http://www.prosnex-tool.com>.

January, 2019

Rasim Murat Aydıncal

LIST OF SYMBOLS

d	: Diameter
L	: Characteristic path length or average path length
n	: Number of nodes
k	: degree
w	: edge weight
w'	: normalized edge weight
C	: Average clustering coefficient
c	: clustering coefficient
PC	: Potential connections
C_B	: Betweenness centrality
G	: Graph
V	: Vertice
E	: Edge

LIST OF ABBREVIATIONS

2D	: Two Dimension
3D	: Three Dimension
PDB	: Protein Data Bank
gRINN	: get Residue Interaction eNergies and Networks
GPU	: Graphics Processing Unit
OpenGL	: Open Graphics Library
WebGL	: Web Graphics Library
1CRN	: Crambin
1OF2	: Human Lymphocyte Antigen HLA-B27
1UBQ	: Ubiquitin
PV	: Protein Viewer
NAPS	: Network Analysis of Protein Structures
RINalyzer	: Residue Interaction Network Analyzer
RING	: Residue Interaction Network Generator
HTML	: Hypertext Markup Language
CSS	: Cascading Style Sheet
JS	: JavaScript
cdjson	: Chemical JSON format
cube	: Gaussian cube format
gro	: Gromacs topology format
mcif, cif	: Crystallographic Information File
mmtf	: Macromolecular Transmission Format
mol2	: Sybyl Mol2 format
pqr	: Very loosely-formatted PDB files

prmtop : Amber topology file, must add coordinates
sdf : MDL MOL format
xyz : XYZ cartesian coordinates file format



LIST OF FIGURES

	Page
Figure 1.1 Common representations of protein structures. Alpha helical bundle (left), Beta-sandwich (middle), Alpha/Beta barrel (right).....	16
Figure 1.2 Two-dimensional protein network example for Protein Databank code 1CRN [73] (Crambin protein) using ProSNEEx.	17
Figure 1.3 Three-dimensional view of the network constructed by using NAPS [27] server for 1CRN [73].....	18
Figure 1.4 An example graph showing the eccentricity	19
Figure 1.5 An example shows the number of actual and potential connections between vertices.....	21
Figure 1.6 An example network shows the degree centralities for each node.....	23
Figure 1.7 Different layouts have been applied to the same network; circular and random	24
Figure 1.8 An example adjacency matrix for the graph shown in Figure 1.9.....	24
Figure 1.9 Graphical representation of an example graph $G=(V, E)$ with vertex set $V=\{1,2,3,4,5,6,7,8,9\}$ and edge set $E=\{\{1,2\}, \{2,3\}, \{2,5\}, \{3,6\}, \{3,7\}, \{4,5\}, \{4,7\}, \{5,7\}, \{6,7\}, \{6,8\}, \{7,8\}, \{8,9\}\}$	25
Figure 1.10 An adjacency list for the graph shown in Figure 1.9	25
Figure 1.11 Properties of the network for 1CRN [73].....	30
Figure 1.12 NAPS general view of shortest path results. Shortest path is computed from 1CRN/A.1 to 1CRN/A.43.....	31
Figure 1.13 RING2 main screen to parameterize options residue interaction network .	32
Figure 1.14 RING2 shows the qualitative and quantitative results for 1CRN [73] protein	32
Figure 2.1 A sample scatter plot.....	41
Figure 2.2 Contact map demonstration.....	42

Figure 2.3 Representation of a distance map	43
Figure 3.1 Select PDB file window	44
Figure 3.2 Chain selection window	45
Figure 3.3 Network settings. There are various network construction options provided for the users.....	45
Figure 3.4 Three main window panels of ProSNEx	47
Figure 3.5 3D structure of the molecule	47
Figure 3.6 Residue interaction network.....	48
Figure 3.7 Topological parameters.....	49
Figure 3.8 Average degree tab and the drop-down menu of this property	50
Figure 3.9 Clustering coefficient plot. The value of each residue is shown and the average value is marked with a red line	51
Figure 3.10 Preferences section.....	51
Figure 3.11 An example of the help mode.....	52
Figure 3.12 Display residue mode.....	53
Figure 3.13 Distance calculation between residues A.PRO267 and A.HIS3 for 1OF2 [131].....	54
Figure 3.14 Display of the neighboring residues mode. Chain A, residue 101 is selected from the network panel, the corresponding neighbors are displayed on the 3D structure ..	55
Figure 3.15 UniProt annotations are marked on the 3D structure and are also listed	55
Figure 3.16 Comparison section, compare metrics button	56
Figure 3.17 Comparison of multiple metrics on the same structure can be done using the ‘compare metrics’ button.....	56
Figure 3.18 Drop-down menu for the 3D structure coloring section	57
Figure 3.19 Coloring of the 3D structure based on b-factor values for 1OF2/A	57

Figure 3.20 Coloring of the 3D structure based on conservation values taken from ConSurf for 1OF2/A.....	58
Figure 3.21 Coloring of the 3D structure based on centrality values for 1OF2/A.....	59
Figure 3.22 Some additional features of the preferences section.....	60
Figure 3.23 Shortest path algorithm drop down menu	60
Figure 3.24 An example shortest path analysis result is displayed for 1OF2, between residues A1 and B84. The total edge weight is given and the path is displayed on both the 3D structure and the network view panels.....	61
Figure 3.25 Display of edge weight option.....	61
Figure 3.26 Display map section.....	62
Figure 3.27 Display of the scatter plot matrix for 1OF2 [131]	62
Figure 3.28 Display of the contact map for 1OF2 [131]	63
Figure 3.29 Display of the distance map for 1OF2 [131].....	64
Figure 3.30 Node centrality analyses section.....	66
Figure 3.31 Degree centrality results of a network constructed for 1OF2/A and 1OF2/B	66
Figure 3.32 Plot of degree centrality values	67
Figure 3.33 Closeness centrality results of a network constructed for chain A of 1OF2... ..	67
Figure 3.34 Plot of closeness centrality results for a network constructed for 1OF2/A.....	68
Figure 3.35 <i>k</i> -clique analysis.....	68
Figure 3.36 Display of <i>k</i> -clique analysis results for a clique size of 6 on 1OF2/A	69

LIST OF TABLES

	Page
Table 1.1 Comparison of the most commonly used visualization tools	27
Table 1.2 The most used web-browser and their versions and years which the WebGL support has become to a standard feature since then.....	27
Table 1.3 GitHub.com statistics for graphical visualization of selected WebGL libraries	29



1.1. Networks of Protein Structures

Proteins are the building blocks of almost all biological processes that constitute life. It is considered that the protein interactions are significant facts of the cell components and cell activity [7]. In order to analyse protein structures, secondary structure and folding properties of proteins are used traditionally. There are many programs that enable the visualization of the protein 3D structures such as RASMOL [8], Chimera [9], Mage [10], VMD [11], PyMOL [12] and Jmol [13]. Such visualization and investigation of protein structures enable us to understand the function of the proteins more deeply. On the other hand, modelling proteins as a network of the non-covalent interactions between amino acid residues is an alternative concept to this traditional approach [14]–[19]. This approach has been used to identify the key residues in folding, classification, packing of proteins and to study the fluctuation dynamics of proteins.

Following this idea, complex systems have been investigated with network (graph) theory [20], particularly with proteins and other biological molecules [16], [19], [21]–[25]. Topology of a network refers to a set of nodes and edges, in general. This topology functions as a potential graph to represent residue–residue interactions, thus it is used to compare various protein conformations of the same molecule [26]. For proteins, structural stability is based upon the complex residue contact network. Nodes are considered as amino acid residues, and edges drawn between the nodes within a cut-off criterion can be used to define the network representation of a protein. Recently, graph visualization of protein interaction methods has become popular in analysing complex protein structures, enlightening the structure–function relationship using different aspects of network topology [27]. Complex systems can be investigated as interaction networks; hence, analysing these networks can describe a system and its related singular elements [28]–[30]. Protein structures are typically perceived as local structure elements that form diverse topologies and folds [30]–[32]. However, protein structures can also be visualized as a network graph where amino acid residues are the nodes and their interactions are the edges [33]. Protein interaction networks can be weighted or unweighted [34].

Network methods have recently been used to investigate different protein aspects such as protein conformation stability [19], protein folding and classification [35], identifying the key residues in protein packing [14], clustering the side-chains [36] and fluctuation dynamics of proteins [25]. Since studies on biochemical properties of proteins are limited, de-novo methods have become crucial in predicting the active sites of proteins [5], [30], [37].

In addition to classical network analysis metrics, one particularly important phenomenon in proteins is signal transmission among the amino acid residues [38]. Algorithms have been developed to estimate the shortest path between two nodes, by considering the edges between these two nodes as unweighted and/or weighted [39]–[41]. Residue centrality concept [42] identifies the evolutionary conserved residues in a protein domain by focusing on the inter-residue connections [43].

The need for visualization and analysis of protein networks that works on the modern data analysis environments such as the web is increasing day by day [15], [23], [44]. A variety of programs are available to create and analyze amino acid networks through graph theory. While Aminonet [45] and GraProStr [46] are only limited to find hubs, clusters and cliques; Con-Struct Map [47], CMWeb [48] can be used to visualize the contact maps. RING [49] enables the users to generate residue-based interaction networks and the results can be visualized by using Cytoscape [50]. The results return by RINalyzer [51] and CytoStruct [52] can be imported to analyze into Cytoscape [50] as well. PyMOL [12] plug-ins which are called xPyder [53] and PyInteraph [54] can be used for the same purposes as well. CSU [55], PSN-Ensemble [25] and NetworkAnalyzer [56] are also used for similar purposes. SeqX [57], Protmap2D [58] and CMview [59] are standalone applications. Recently developed NAPS [27] can be used to visualize protein network and to analyze network parameters. Protein Contact Atlas [60] enables users to generate non-covalent interaction network by offering various visualization options for the conformations deposited at PDB [1].

Despite the functionalities of desktop applications that are increasingly being ported to web, there are no efficient, user-friendly web-based solutions for the representation and analysis of biological, yet. They provide narrow functionality, mostly limited to network visualization and basic analysis options.

1.2. Protein Interaction Networks and Related Definitions

As the biological networks have been proved to explain the interaction between the subunits of biological molecules, they have started to gain more interest in the recent years [61]–[65]. Recent innovative technologies [66] reveal experimentally big-data related to protein interactions. Protein-protein interactions, as a specific type of biological networks, especially focus on proteins in the cell during the analysis. Nodes indicate proteins, while edges describe the pairwise interactions in a typical protein-protein interaction network. Proteins are synthesized in cells into distinct complexes in order to have a particular role [67], [68]. Proteins often interact to participate in the same biological activity and can be associated with highly interrelated tasks [69]. Protein with significant biological definition can be a part of a complex protein system or can be a portion of any function. Basically, a protein complex can be defined as various protein structures, which become to be interrelated with each other; whereas functional modules explain that a special function is cooperated with more than one protein that have come together. Unlike protein complexes, proteins in a functional module do not need to have a spatial position at the same time and place [70].

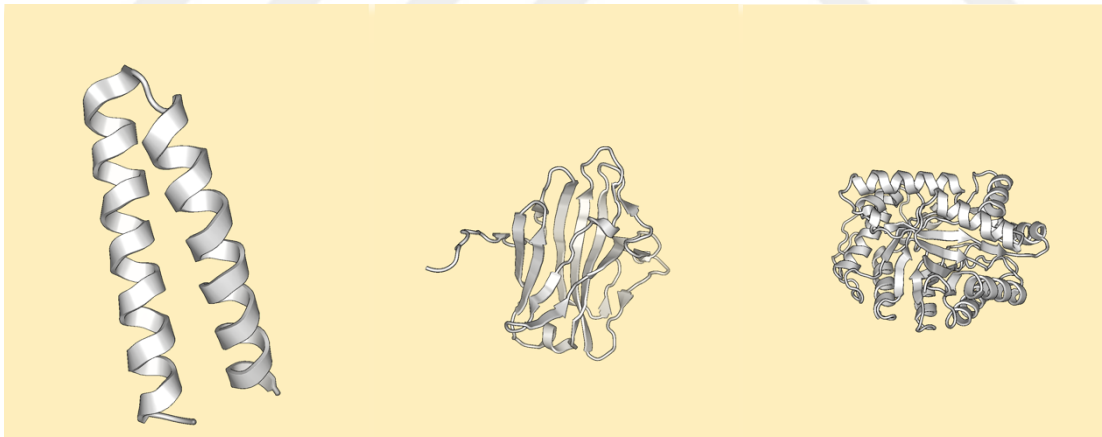


Figure 1.1 Common representations of protein structures. Alpha helical bundle (left), Beta-sandwich (middle), Alpha/Beta barrel (right).

Figure 1.1 represents the common view of protein structural classes. Secondary structures are established into 3D settlements which may be classified into different fold character which are called super secondary structures or structural motifs such as up-and-down alpha-helical bundle, a mixed Alpha/Beta barrel, and etc. Two-dimensional (2D) graph or three-dimensional (3D) network occurrences can be performed by measuring the interactions of amino acids during the translation of that structure [44]. Measuring the

interactions of the amino acids in space is a crucial point. This measure can be the number of residues in the primary structure of connected couple, or the length between the collaborated residues in a 3D conformation. Both types of metrics depend on short or long range interactions. Short range interaction is referred to the connections among the residues in secondary structures. It is commonly being assumed that Hydrogen bonds, for instance, in an alpha-helix are clear identifiers. Long range interactions can be basically defined as the distance between interacting residue couples spotted in disparate secondary structures [44]. The location of interacting residues in the primary structures may or may not be considered, and the distances of interacting pairs of residues differ between 4 and 10 Å, for example, in the tertiary structure in creating of the networks [14], [15], [18], [19], [35], [71], [72]. Interaction can be measured by considering all heavy atoms between residues or by using a representative atom for residues, such as Carbon-alpha or Carbon-beta.

An example of conversion from a protein structure to 2D or 3D network graph is shown in Figures 1.2 and 1.3.

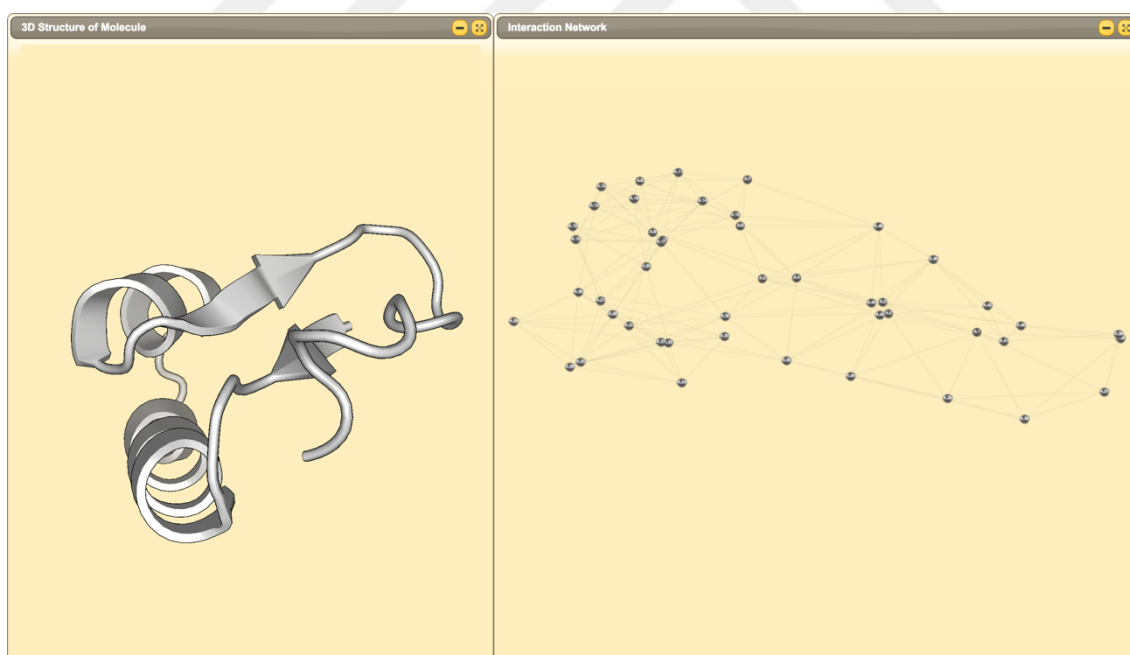


Figure 1.2 Two-dimensional protein network example for Protein Databank code 1CRN [73] (Crambin protein) using ProSNEx.

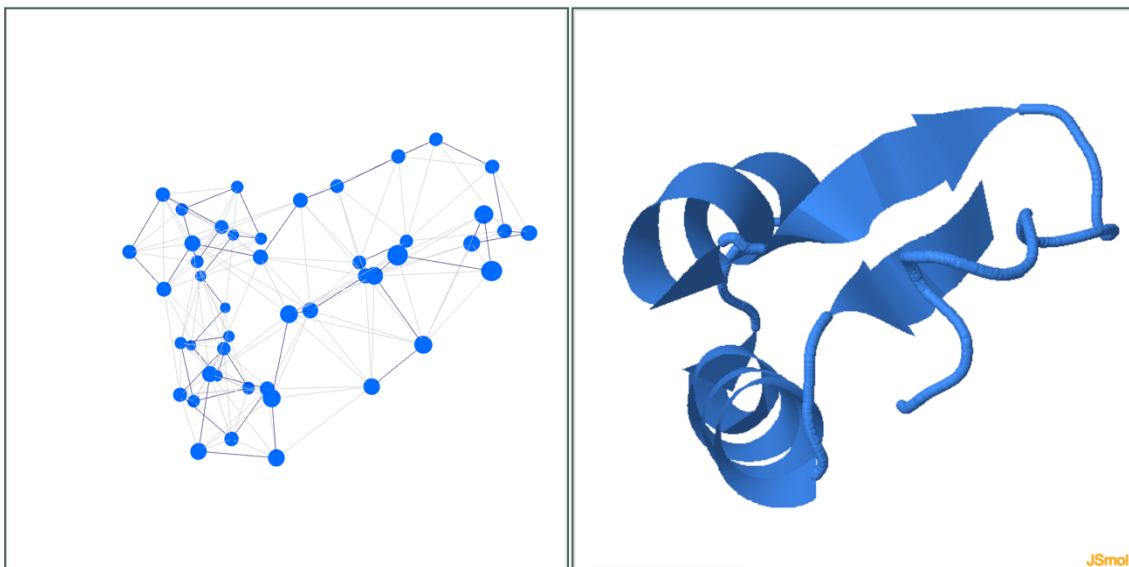


Figure 1.3 Three-dimensional view of the network constructed by using NAPS [27] server for 1CRN [73].

There are many algorithms and computer software tools that deal with these distance calculations in order to create, visualize and analyse a protein residue interaction network. Cytoscape [50], Pajek [74], RING2 [43] and RINalyzer [51] are the most commonly used tools in the field of network science for describing the nodes and edges. Conversion from a protein structure to an interaction network is strictly based on the mathematical rules and assumptions in protein network science.

1.3. Network Parameters

There are various parameters in the area of network studies. Some common parameters will be explained in this section.

Unweighted network is a network type where each edge is considered to be equal. This may result in bringing out alternative paths with equal path lengths.

On the other hand, *weighted network* stands for a network where the edges between the vertices have weights.

The *shortest path* is the total edge length which is called *distance* d_{ij} from any two vertices n_i, n_j . There may exist more than one shortest path from n_i to n_j in an unweighted network.

The *Dijkstra* [40] and the *Floyd-Warshall* [39] are the most commonly used algorithms in the shortest path analysis [75]. Dijkstra's shortest path algorithm presents the minimum length of path from n_i to n_j vertices of the network. Since calculating all the paths and

selecting the shortest one is actually time-consuming, it returns only a single shortest path from source to target nodes, so as to work efficiently [76]. The Floyd-Warshall algorithm is the suitable option to calculate the path length of all pairs of nodes. Dijkstra and Floyd-Warshall algorithms are appropriate for weighted networks. However, if the network has negative edge weights, then Dijkstra does not work. On the other hand, *Bellman-Ford* algorithm [41] is compatible to work with negative edge weights as well.

The *eccentricity* of a node in a graph G , is considered to be the maximum distance from n_i to other vertices, as shown in Figure 1.4.

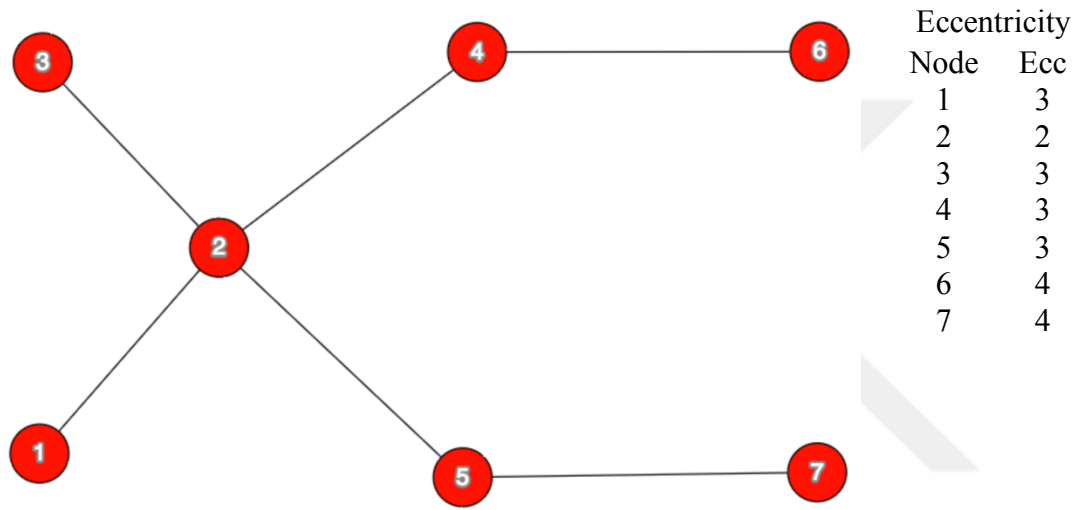


Figure 1.4 An example graph showing the eccentricity.

The *diameter* refers to the maximum distance between any two pair of nodes in a network.

$$\text{diameter } d_m = \max(d_{ij}) \quad (1.1)$$

In other words, the *maximum eccentricity* is equal to diameter. Generally, the vertex with higher eccentricity values is located at the sides of the graph; while the nodes with low eccentricity are the most central vertices in the graph.

The *radius* of a graph refers to the *minimum eccentricity* value.

The *characteristic path length* or *average path length* (L) of a network is the shortest path length between two vertices averaged over all pairs of vertices. It is calculated by the following formula:

$$L = \sum_{i,j \in V} \frac{d(i,j)}{n(n-1)} \quad (1.2)$$

If a network has high average path length, than it is considered to be in linear form; while low average path length deals with compact networks.

The *clustering coefficient* is a critical metric for describing the characteristic structure of a protein. For unweighted network, the clustering of a node u is the fraction of possible triangles through that node, shown in the Formula 1.3:

$$c_u = \frac{2T(u)}{\deg(u)(\deg(u)-1)} \quad (1.3)$$

where $T(u)$ is the number of triangles through node u ; and $\deg(u)$ is the degree of u .

For weighted graph, it is defined as the geometric average of the sub-graph edge weights. It can be calculated with the following Formula 1.4 [77]:

$$c_u = \frac{1}{\deg(u)(\deg(u)-1)} \sum_{uv} (w'_{uv} w'_{uw} w'_{vw})^{1/3} \quad (1.4)$$

where the edge weights w'_{uv} are normalized by the maximum weight in the network. The value of c_u is assigned to 0 if $\deg(u) < 2$.

$$w'_{uv} = \frac{w_{uv}}{\max(2)} \quad (1.5)$$

The *average clustering coefficient*, C is the average of all clustering coefficients of all nodes, c_u [77] :

$$C = \frac{1}{n} \sum_{u \in G} c_u \quad (1.6)$$

where n is the total number of vertices in a graph.

Network density is the ratio of actual edges to potential edges (connections) between vertices. Potential connections (PC) can be calculated by the following formula:

$$PC = \frac{n(n-1)}{2} \quad (1.7)$$

where n is the number of nodes in a network. An example is shown in Figure 1.5.

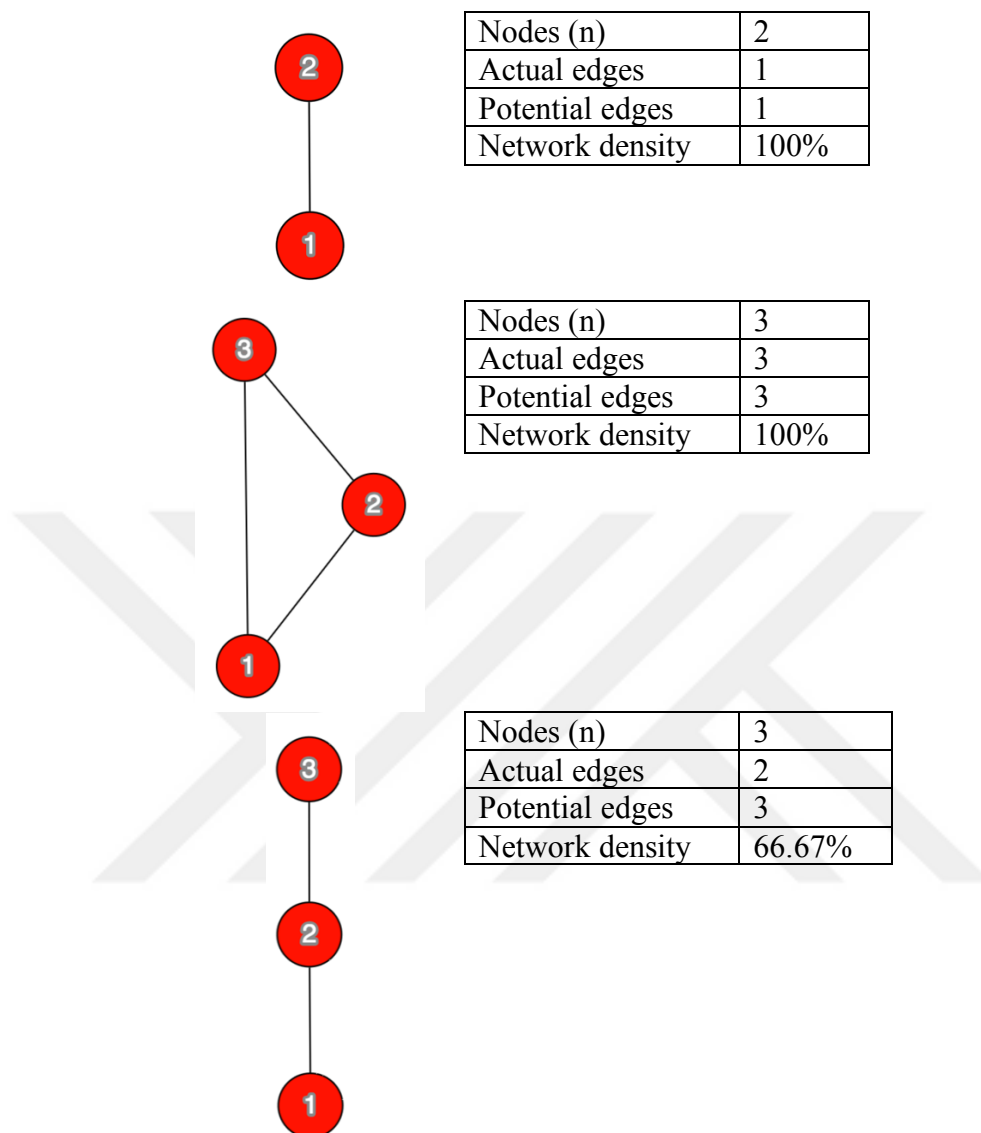


Figure 1.5 An example shows the number of actual and potential connections between vertices. If a network has isolated vertices without any edges between any pair of nodes, the density returns with 0. The *network density* is generally used to compare network graphs at a first glance. It depends on occupying edges and how densely they are found in the interaction network.

Betweenness centrality is a parameter to identify the relative importance of a node and how many times a node is passed in a route of shortest path from any pairs of vertices. The value is normalized by the total number of vertex couples. If a node's betweenness value is high, this means that this node is central in transmission a signal in a pathway. This metric may represent valuable information about communication paths in a structure.

Recent drug-design studies are interested in the residues with high betweenness centrality values to simulate docking of a designed ligand to a protein [78]. Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that include this vertex:

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \quad (1.7)$$

where V represents the set of vertices; $\sigma(s,t)$ is total number of shortest paths; $\sigma(s,t|v)$ is the number of shortest paths that include vertex v , except s,t . Edge weights must be a positive number except 0 to prevent having infinite number of equal path lengths in the calculation of betweenness centrality.

Closeness centrality of a node is a measure to clarify its closeness to alternative nodes in a network. It compares and rates a node by calculating all-pair shortest paths in a graph with shortest path summation of this node. Closeness centrality informs the impact of a node based on its leadership in an undirected network.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)} \quad (1.8)$$

where $d(v,u)$ is the shortest path length from source node v to target u , and n is the count of nodes in an interaction network.

The *residue centrality* is the efficacy of a node removal in average path length, L in a network [42].

Degree centrality signifies an importance of a node based on its direct connections to the other nodes in a graph. It is used to find influential nodes by having lots of connections, or nodes which are disposed to stay closer to the most of the nodes. Figure 1.6 shows an example for degree centrality.

The *average degree* in an undirected graph is the average score of edges per node and it is calculated by:

$$\langle k \rangle = \frac{2E}{n} \quad (1.9)$$

where k is the degree, E is the total edges, and n is the number of nodes in an undirected network. If the degree centrality of a node is higher than the average degree, then this node is considered to be a *hub node*. Hub nodes which are considered to have major feature for topology of network arise through scale-free characteristics of the topology.

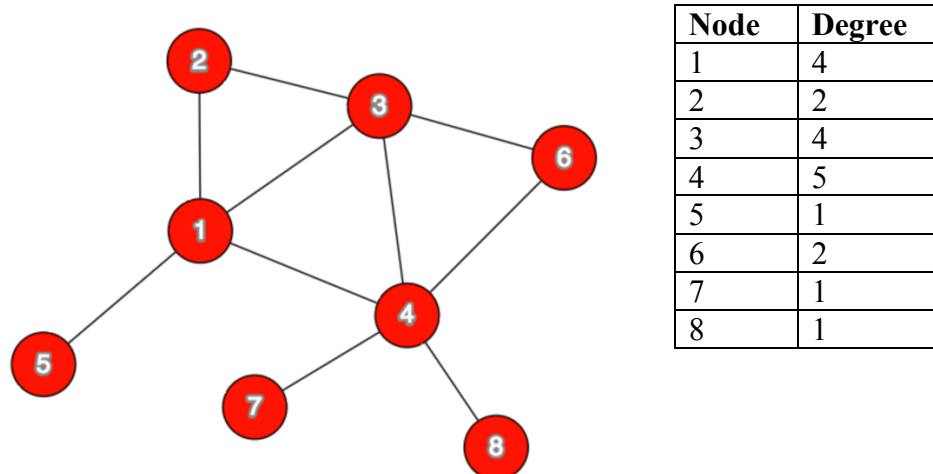


Figure 1.6 An example network shows the degree centralities for each node.

Eigenvector centrality measures the effect of a node on the number of edges to other nodes on the network by considering the number of neighbours of the other nodes. This metric can determine the impact of a node on the entire network by dealing with the extended connections of a vertex, not only directly linked connections. A node with high eigenvector centrality value implies that this node has connections which have high values. All the eigenvector centrality scores have positive values.

Clique is a complete subgraph or subset of vertices that are pair-wise adjacent in an undirected graph.

1.4. Visualization of Biological Networks

A network is defined by the representation of a set of nodes and edges between them in a graph, such as a protein interaction network. In this case, proteins or residues stand for the nodes, and the edges between nodes denote the interactions among these nodes.

Networks are designed by mathematical graphs that are made up with vertices which are connected by edges. The graph theory [79] is a term which is the major interest of computer sciences, biology, mathematics and social sciences. A graph contains a set of distinct vertices, which are called nodes and edges shown as $G=(V, E)$ where V is the set of nodes, and E is the set of edges. A connection between vertices, u and v , is designated by $\{u, v\}$. A graph can be: *directed*, *undirected*, or *mixed*. In a directed graph, all the edges have a specific direction, while an undirected graph has bidirectional connections. The *layout* refers to the appearance of a graph, which shows the location of the vertices and the edges in a graph. Random layout algorithm [80], circular layout algorithm [81],

hierarchical layout algorithm [82], Fruchterman & Reingold and Eades Peter algorithm [83], [84], Kamada-Kawai algorithm [85], tree layout algorithm [86] and compound spring embedder layout algorithm [87] are often cited layout strategies. Figure 1.7 shows two different sample layouts for the same network.

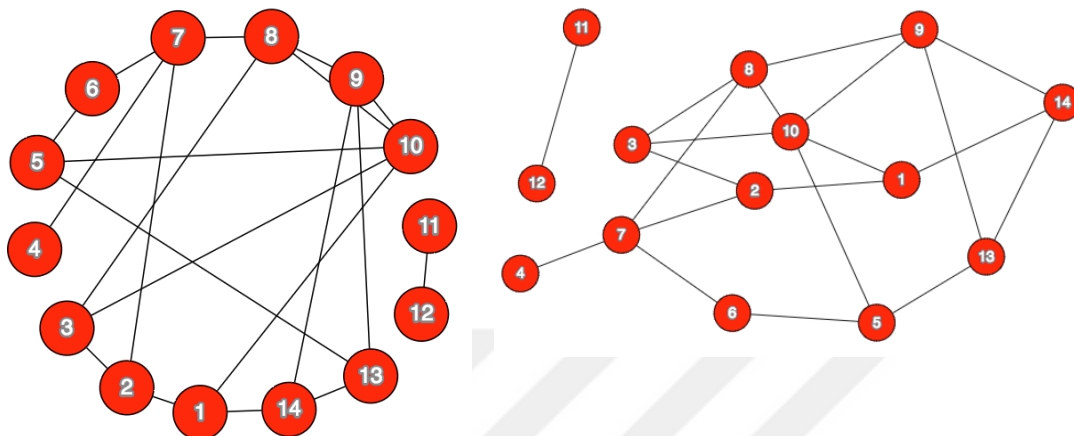


Figure 1.7 Different layouts have been applied to the same network; circular and random.

Graphs are introduced to computers by adjacency matrix and adjacency list. Adjacency of vertex pairs is expressed in a boolean number with the elements of a $(n \times n)$ adjacency matrix. Although adjacency matrices are easily applied and are very common in the representation of biological networks, adjacency list representation is more favourable in respect to the memory usage and run-time performance concerns [76]. Figure 1.8 and Figure 1.10 represent adjacency matrix and adjacency list of the network demonstrated in Figure 1.9.

vertex	1	2	3	4	5	6	7	8	9
1	0	1	1	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0	0
3	1	1	0	0	0	1	1	0	0
4	0	0	0	0	1	0	1	0	0
5	0	1	0	1	0	0	1	0	0
6	0	0	1	0	0	0	1	1	0
7	0	0	1	1	1	1	0	1	0
8	0	0	0	0	0	1	1	0	1
9	0	0	0	0	0	0	0	1	0

Figure 1.8 An example adjacency matrix for the graph shown in Figure 1.9.

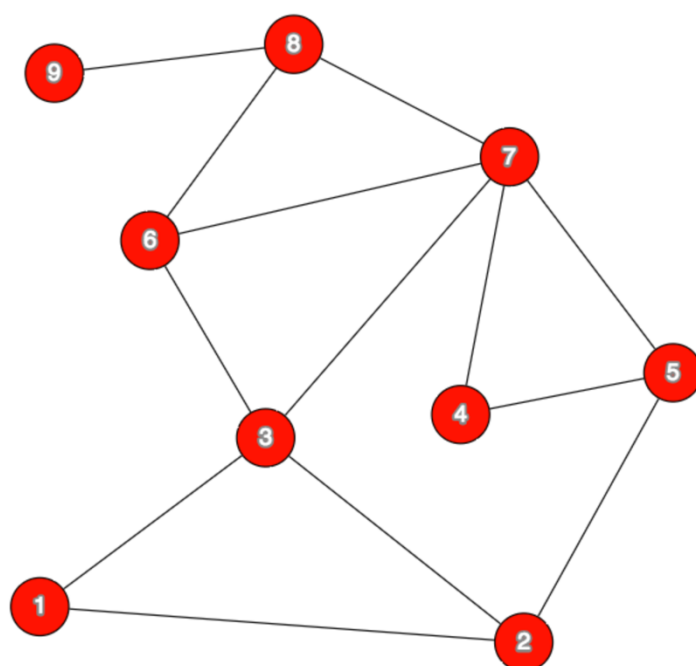


Figure 1.9 Graphical representation of an example graph $G=(V, E)$ with vertex set $V=\{1,2,3,4,5,6,7,8,9\}$ and edge set $E=\{\{1,2\}, \{2,3\}, \{2,5\}, \{3,6\}, \{3,7\}, \{4,5\}, \{4,7\}, \{5,7\}, \{6,7\}, \{6,8\}, \{7,8\}, \{8,9\}\}$.

$L_1: (\{1,2\}, \{1,3\})$
 $L_2: (\{2,1\}, \{2,3\}, \{2,5\})$
 $L_3: (\{3,1\}, \{3,2\}, \{3,6\}, \{3,7\})$
 $L_4: (\{4,5\}, \{4,7\})$
 $L_5: (\{5,2\}, \{5,4\}, \{5,7\})$
 $L_6: (\{6,3\}, \{6,7\}, \{6,8\})$
 $L_7: (\{7,4\}, \{7,5\}, \{7,6\}, \{7,8\})$
 $L_8: (\{8,6\}, \{8,7\}, \{8,9\})$
 $L_9: (\{9,8\})$

Figure 1.10 An adjacency list for the graph shown in Figure 1.9.

1.5. Computational Tools

Gradually evolved bioinformatics applications revealed big data at protein level [88]. Innovative computational tools and methods are crucially required to view, manipulate and to get a molecular level understanding of this biological data. In this section, current network visualization tools of protein systems will be discussed.

1.5.1. Network visualization tools

In order to examine the biological systems, pairwise interactions of proteins are critical. Visualization of the complicated information such as interaction networks, or protein complex structures by a graph is believed to be the most practical approach in structural

biology. Current applications that represent the visualization of protein interactions have miscellaneous functions and features. Among these tools, Cytoscape [50] is the most widely used application. These tools have been explained and discussed in previous reviews [89]–[92]. Even though visualizing protein residue contact networks is a functional procedure, there are several challenging issues:

- Need of a highly specific hardware to manipulate numerous nodes and to maintain lots of edges in a graph to prevent rendering problems of graph images.
- Difficulty in association of corresponding characteristics of interaction data to visualization procedures.
- Presence of various data formats for protein interaction networks.

When designing a visualization tool, some fundamental requirements of the users should be taken into account such as short return time, comprehensibility of interface, dynamic data architecture, possibility to choose alternative graph layouts and alternative graphical rendering methods including both 2D and 3D as well as availability of a graphical interface.

1.5.1.1. Cytoscape

Among the analysing tools, Cytoscape [50] is the most widely used tool as shown in Table 1.1. Users commonly use this tool in order to have significant implications about protein interactions [91], [93]. Cytoscape is freely available at <http://cytoscape.org>. It has many pre-defined functions and algorithms so that the users can analyse topological parameters of networks. Grouping proteins according to their annotations and attributes is one of the most favourable features of Cytoscape. This tool lets users to integrate required packages from its repository which contains many plug-ins.

Max Franz et al. [94] have published a JavaScript library for Cytoscape features to be used on the modern web browsers. Since Cytoscape.js has no dependencies and encloses diverse environments such as jQuery, it can easily be imported into the HTML5 based applications. On the other hand, it has a disadvantage of lacking a 3D layout.

Table 1.1 Comparison of the most commonly used visualization tools

Tool Name	Total Citations	Data Retrieved	Freely Available	2D/3D	Plugin Support	Reference
Cytoscape	14652	24/09/2018	Free	2D	Yes	[50], [95]
Gephi	3847	24/09/2018				[96]
Pajek	714	24/09/2018	Free	Both	No	[74]
Medusa	177	24/09/2018	Free	2D	Yes	[97]
NAViGaTOR	168	24/09/2018	Free	Both	No	[99]
BioLayout	174	24/09/2018	Free	Both	No	[98]
Arena3D	90	24/09/2018				[100]

On the other hand, it has a disadvantage of lacking a 3D layout.

Desktop applications including VMD [11], PyMOL [12] and Chimera [9], etc. could use GPU-based hardware acceleration through OpenGL to render 2D or 3D vector graphics. In order to display 3D graphics on web-browsers, some plug-ins are required which are called java applet, for instance: Jmol [13] and OpenAstex Viewer [101]. On the other hand, it is not a good idea to use java applets for the security concerns, having annoying prompts and slow work. For this reason, Jmol is ported to JavaScript with Jsmol library. Nowadays, modern web browsers can operate with hardware-accelerated graphics through the WebGL.

Table 1.2 The most common web-browsers, their versions and years which the WebGL support feature has become to a standard feature since then

Web Browser	Version	Since Year
Firefox	5	2010
Google Chrome	9	2011
Safari	5.1	2011
Opera	12	2012
Internet Explorer	11	2013

The web-based 3D molecule viewer studies started with GLMol [102], which is still slow. Features of common WebGL based molecule viewers are compared on Table B.1 in Appendix B.

1.5.1.2. Other tools

Development of high-throughput technologies brings out tremendous biological data. This brings along tools to view and manipulate this immense data. The most important criterion in choosing a tool to computationally analyze biological data is to be aware of

the capabilities and limitations of these tools. In this section, several visualization tools are given by their pros and cons.

Gephi [96] lets the user to manipulate the source code since it is developed at an open-source platform. This tool supports to view both directed and undirected graphs with real-time visualization up to 100,000 nodes and 1,000,000 edges. Users can change the layout while running by using the layout palette. The framework of this tool enables the user to measure betweenness centrality, closeness centrality, clustering coefficient, diameter and shortest path analysis. It supports 3D rendering on interaction networks. However, Gephi does not support retrieving data from protein databases including IntAct [103], STRING [104], DIP [105], Reactome [106] and I2D [107]. This tool is not specifically designed for analyzing protein-protein interactions.

Pajek [74] provides efficient tools for analysis of large networks including protein interactions. Pajek enables users to cluster the networks. Pajek only works with its own data format which results in a limited number of users.

Medusa [97] is developed under Java platform to visualize and cluster the protein-protein interaction networks. It supports 2D rendering and integration to STRING [104] database. Without installation, Medusa can be run at any operating system with Java. In order to visualize the network on the web, it has a Java applet to show and cluster the interaction network basically. However, it only works with small number of nodes, and basic Java scripting is a necessity to use this application.

BioLayout [98] supports both 2D and 3D visualization of biological networks. It maintains large weighted networks with a simple input file. However, it cannot integrate to interaction databases. User settings cannot be saved such as node color. BioLayout was free on the internet but it is no longer supported.

NAViGaTOR [99] is suitable for large data-sets by importing data from protein interaction databases including I2D [107]. This tool supports both 2D and 3D view. It has a disadvantage of lacking any plug-in for structural analysis as a close-source application.

Arena3D [100] is a 3D visualization tool for biological interactions of proteins and signaling pathways. The tool supports to find the similarities of the network graphs by considering their topological parameters. It is developed under the Java platform and free

for non-commercial uses. There is no feature to change the default options for node size, color and etc.

iGraph [108] is a free network analysis package that can be programmed in R, Python and C/C++. This package is suitable for simple and complex network analysis by supporting graph visualization and centrality methods.

NetLogo [109] is used to model network dynamics with its very limited features. It can only measure the clustering coefficient and the average path length. This is not designed for especially biological networks.

NodeXL [110] is a Microsoft Excel 2007/2010/2013/2016 template for the exploration of network graphs. NodeXL can measure the basic topological parameters including betweenness centrality, closeness centrality, eigenvector centrality, clustering coefficient and graph density.

In addition to these applications or packages, there also exist various JS libraries used in visualizing 2D or 3D graphs through WebGL, which are vis.js [111], Ngraph [112], VivaGraphJS [113], Dagre [114], D³ [115], Springy [116], Alchemy [117], Sigma [118], Dracula [119], Cytoscape.js [94] and JSNetworkX [120]. Statistical comparisons of these libraries are given at Table 1.3.

Table 1.3 GitHub.com statistics for graphical visualization of selected WebGL libraries

Library	GitHub Stars	GitHub Forks	Created Year	Reported Issue
D ³	79052	19784	2010	6
Sigma	8146	1273	2011	443
vis.js	7132	1338	2013	714
Cytoscape.js	4708	847	2011	104
VivaGraphJS	2673	346	2011	77
Dagre	1817	261	2012	83
Springy	1600	238	2010	18
Ngraph	785	99	2013	19
Dracula	654	114	2010	14
JSNetworkX	504	142	2012	35
Alchemy	435	204	2013	67

1.5.2. Tools for topological analysis of protein interactions

In this section, an overview of some frequently used web servers or tools which have been developed for a wide audience of users and offering numerous metrics and features of fundamental network analysis, will be given. The raw biological data or primary data are

processed by these applications in order to produce significant assumptions through visualization. Each has its own restraints and distinctive features such as efficient use and user friendly layout design.

1.5.2.1. NAPS: Network Analysis of Protein Structures

NAPS is a web server application used for analysing protein structure networks developed by Chakrabarty and Parekh [27]. NAPS provides opportunities for users to make analysis on single/multiple chains and protein complexes. Atom type and residue cut-off for constructing networks is optional. After a network is created by the selected parameters, the measurable metrics are shown such as the number of nodes and edges in the network, diameter, radius, average degree, average shortest path and average clustering coefficient. Users are able to view the 3D network, and some quantitative metrics are shown for further analysis by its neighbourhood measures of a node such as centrality, k -clique, etc. Based on functional analysis, shortest paths between any two pair of vertices can be identified and visualized on both 3D network and structure, so as to discuss and recognize the functional residues, to understand protein folding mechanism and signalling pathways between proteins or protein residues. This application can be accessed on the URL <http://bioinf.iit.ac.in/NAPS/>. Figure 1.11 and Figure 1.12 shows an example view for NAPS.

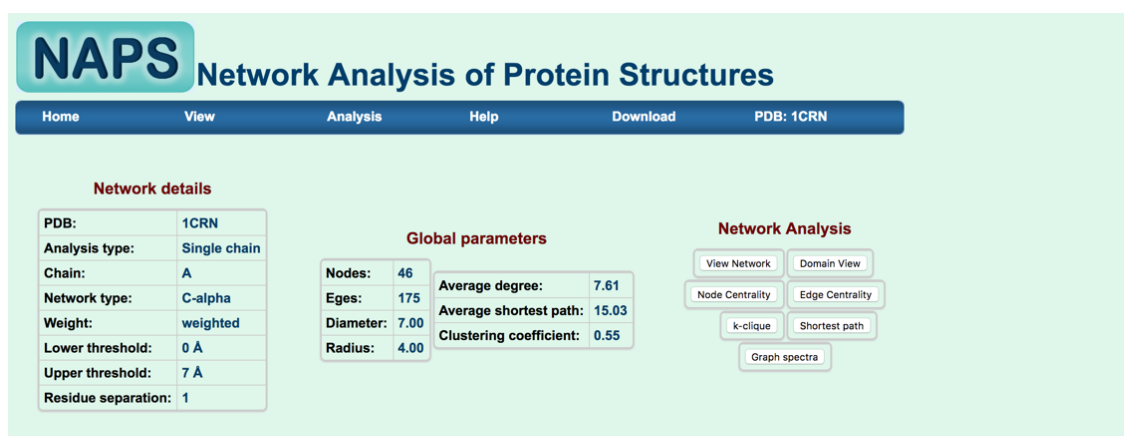


Figure 1.11 Properties of the network for 1CRN [73].

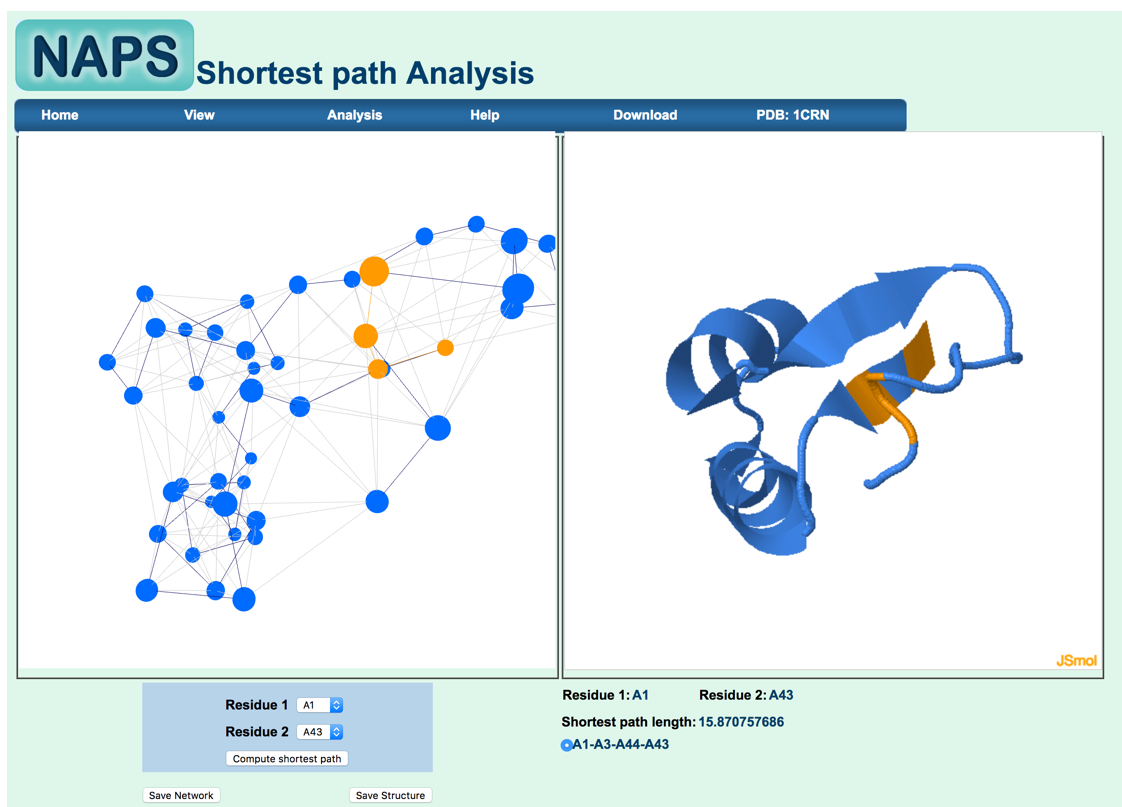


Figure 1.12 NAPS general view of shortest path results. Shortest path is computed from 1CRN/A.1 to 1CRN/A.43.

1.5.2.2. RING2: Residue Interaction Network Generator

RING2 [43] creates detailed networks by considering residue interactions and multiple chain contacts involving atoms of solvent and ligands. RING2 can customize the user selected distance cut-off parameter by reparameterization of calculations. NMR model can optionally be selected by the user. It uses Cytoscape.js [94] library to show the interaction network on the web browser. Interaction network can be analysed with Cytoscape [50], [95], while structure contacts at the atomic scale can be visualized with RING-Viz plugin for PyMOL [12]. The RING2 web server is available at URL <http://protein.bio.unipd.it/ring/>. Figures 1.13 and Figure 1.14 represent some example views of RING2 server usage.


[About](#) [Help](#) [Tutorial](#) [News](#) [Contact](#)

New Version

 RING v2.0.1
 05/04/2016 [More Info](#)

Start

To start using RING just input your **PDB file** or **PDB ID**.
 To retrieve a job use the **Session ID** search box (top-right corner)

Contact Info

For questions and/or comments, please use the [contact form](#).

Integrated resources

RING uses resources from the following projects



Citing RING

Piovesan D., Minervini G., Tosatto S.C.E.
The RING 2.0 web server for high quality residue interaction networks.
 (2016) Nucleic Acids Research, 44 (W1), pp. W367-74. [PubMed](#) [DOI](#)

The **Residue Interaction Network Generator (RING)** identifies all types of **non-covalent** interactions at atomic level in a **protein structure (PDB)**. The output can be visualized in two different ways.

- **Interaction network**, with this web server or with Cytoscape/RINalyzer/StructureViz
- **Structure contacts**, with the [RING-Viz script](#) for Pymol

The tool can be run directly from [this web server](#) or locally on your computer by downloading the [RING-2.0 software](#). Visit the [help](#) page for information about this web server and output formats. The [tutorial](#) for help using RING-Viz script, Cytoscape and RINalyzer/StructureViz plugins. See [about](#) for details about the RING method.

Input

PDB ID

Provide a PDB identifier...
Example 1, example 2

PDB File
 No file chosen
...or your own PDB file

Options

Distance thresholds
☐ Relaxed ☒ Strict ☐ Manual

Chain

Model (only NMR)

Sequence separation

Network Policy
☒ Closest ☐ Lollipop ☐ C-alpha ☐ C-beta

Interaction type
☐ One ☒ Multiple ☐ All ☐ No specific

☐ Skip hetero atoms and ligands
☒ Skip water molecules
☐ Skip FRST and TAP energy calculation
☐ Calculate mutual information (slow!)

Figure 1.13 RING2 main screen to parameterize options residue interaction network.


[About](#) [Help](#) [Tutorial](#) [News](#) [Contact](#)

Session ID: 5ba77216b7fb16a58250b8 [Network \(xml\)](#) [Network \(txt\)](#) [PDB](#) [Figure \(SVG\)](#)

Please save the session ID or the page link to come back later on your results

1CRN.pdb

Network policy **Ca**
 Interaction type **Multiple**
 Sequence separation **3**

Chain **all**
 Nodes **40**
 Edges **75**

Node color
 Type of residue

Repulsion(130)
 Gravity(0.05)
 Attraction(0.2)

ARG LYS CYS ILE LEU
 MET PHE PRO TRP TYR
 VAL ASN GLN HIS ALA
 GLY SER THR ASP GLU
 LIG

HBOND VDW SSBOND



Figure 1.14 RING2 shows the qualitative and quantitative results for 1CRN [73] protein.

32

1.5.2.3. NetworkX

NetworkX [121] is a complete package, developed under Python scripting language, and can be used to construct, reshape and explore the structure of complex networks. Representation of different kind of networks or both directed and undirected graphs are processed by the core framework of NetworkX. As it is an open-source application, all the built-in functions and algorithms can easily be manipulated at any time. This package gives the optimal opportunity for different disciplines to fit the data to investigate the topological features of the networks including shortest path analysis, betweenness centrality, degree centrality, eigenvector centrality, k -clique analysis, graph clustering and clustering coefficient, etc. NetworkX is compatible with different types of data formats and it can convert the input data to the desired graph format. The stable NetworkX package can be downloaded at URL [<https://networkx.github.io/>](https://networkx.github.io/).

In recent years, a study called JSNetworkX [120] by Felix Kling has attempted to aid the popular features of NetworkX packages to be ported into JavaScript language to enable the usage of these features on the web browsers without downloading the stand-alone application. JSNetworkX library is available at [<http://jsnetworkx.org>](http://jsnetworkx.org).

1.5.2.4. RINalyzer: Residue Interaction Network Analyzer

RINalyzer [51] is a module that supports functional structural analysis components in order to analyse residue interaction networks by facilitating 2D and 3D views. A Cytoscape plugin, structureViz2 [122], is also developed to explore structural data and to visualize with flexible layouts. It can be accessed at [<https://rinalyzer.de/index.php>](https://rinalyzer.de/index.php).

1.6. Motivation of Developing ProSNE_x

Following the previous studies in the field, here is a sum of the features that should be ideally present in a protein structure network construction and analysis tool:

- The developed program needs to automatically fetch conformation data on the Protein Data Bank [1] or should be accepting custom PDB files to be uploaded by the user.
- It should be based on the structural parameters ($C\alpha$, $C\beta$, atom-pair-contact, centroid, etc.) when creating a network. In addition to this, the tool should be able

to create a network by using the information in different formats that can be uploaded by the user. In order to create a weighted network, the tool should be offering normalization options if the user prefers to use correlation data, interaction energy (Molecular Dynamics output etc.) data or similar data as edge weight. This will make the data more functional.

- After the network is created, the features of this network which are the number of nodes, edges, diameter, radius, average degree, average path length, density and clustering coefficient should be reported. Preferably, the structure of network should be visualized both in 2D and 3D structures.
- The algorithms to find the shortest path should be also offered to the users, especially Dijkstra [40], Floyd-Warshall [39] or Bellman-Ford [41] algorithms. The shortest path between any selected two nodes should be shown both as a list, and on 3D and 2D structures.
- Edge weight of any two nodes should be calculated. Contact map and distance map visualization options should be present.
- Node centrality, degree centrality, closeness centrality and betweenness centrality values should be calculated and visualized both in a tabular format and in a graph. In addition to this, the user should be offered to download these results. Network and the 3D conformation should be colored according to the selected centrality values.
- k -clique analysis should be present.

Although many of the programs discussed in the previous section are able to perform analysis on network topology, almost none of them have all of these features alone. The tool developed in this study comprises all these analysis options alone. Additionally, it works on the web environment without installing any dependencies. Unlike others, this web tool also offers:

- An option to use of interaction data specified by the user by providing normalization options.
- Usage of NMA [3] and interaction energies return by gRINN [2].
- Visualization of the results both on the 2D network and 3D structure.

- An option to color the 3D structure based on the ConSurf scores [5] which are automatically fetched from the internet (unless the user uploads a custom PDB file).
- Coloring the structure based on node centrality, degree centrality, closeness centrality and betweenness centrality.
- Demonstration of the results by a table or a graph, and a download option for all the calculated metrics
- Showing the measurements for a specific residue when mouse hovers on it. This will enable the users to see the results (node analysis results, ConSurf score, b-factor value etc.) when moving on the structure without opening a separate window for this purpose. Thus, this tool will be able to clearly demonstrate the role of each amino acid.
- A mode to compare different metrics on the same residue.
- A list to show all metrics sorted by residue or the descending value, and an interactive graph.
- A feature to show the locations synchronously when moving on 3D conformation or network view panels.
- A list for the annotations which will be extracted from UniProt [6] database. This will enable the users to ease their analysis. The program will automatically obtain the natural variant data for the protein and will mark these varying regions on the 3D conformation. The analysis of these annotation data can be detected more easily to identify the important amino acid regions.

As a result, in computational biology, although there is a rapid increase in the number of web applications and functional web tools that create protein structure network, there is no web-based application where all the features mentioned above are combined together.

The most important originality of this study will be an integration of all these approaches that allow the analysis to be made on the web without installing any program. This tool will allow to examine the residues having a key role in signal transduction and can be used in allosteric studies as well.

1.7. Aim of the Study

To determine the function of proteins, understanding the interactions between proteins is crucial. The visualization of protein networks on 3D conformations of biomolecules is essential since network investigation is common for identifying the function and dynamics of biological macromolecules. The function of a protein depends only on a few amino acids; however, the protein structure is rather complex. A large fraction of the protein is expected to participate to its function. Thus, it is proper to consider the protein structures as a completely connected network and then deduce its related properties, which are related to the global network features.

Network-based models of the proteins are hot tools utilized to determine dynamic properties corresponding to the folded structure. They comprise of all topological and geometric computational applications from proteins as directly interacting nodes. Coarse grained approaches efficiently guess alternative conformations because of constitutional integrity in the protein structure. Such a collectivity is moderated by topological characteristics that also define the neighborhood structure. Geometry specifies the comprehensive information of force balance to network models, launching the local shape of the energy landscape. Network view is useful in understanding long-range links among protein residues, contact map highlights the secondary structure arrangement. It supports different analyses and visualization metrics for identifying functional residues including potential targets for therapeutic intervention, providing insight into the mechanisms of protein folding and protein-protein interactions as well as for understanding the communication within and between proteins.

Here in this study, we have developed a web-based application called ProSNEx (Protein Structure Network Explorer) which provides a novel and enhanced analysis of protein structures using network formalism.

This newly developed tool supports the output of a model structure obtained from any molecular modeling program in PDB format, or the files supplied from the Protein Data Bank [1]. To analyze the 3D topology of protein structure, nodes are added with their 3D coordinates of the representative atoms in each network type using the information in the PDB file. This tool supports single chain as well as multimeric proteins and protein complexes. Depending on the type of analysis to be performed, the user can select

chain(s), network type and a distance threshold for network construction. Weighted networks can also be constructed by using either the residue-residue interaction energies in the format returned by gRINN [2] or the dynamical cross correlations from Normal Mode Analysis (NMA) [3] of the protein structure which are automatically retrieved from the PDBj database [4].

Upon construction of the network, the tool offers various types of comparative analysis alternatives to guide the user. Common network analyses metrics (such as global network properties, node degrees, betweenness and closeness centralities) are available offering both visual representations as well as downloadable data files. Shortest path analysis and k-cliques calculation can be performed as well.

The tool is designed to be very user friendly and easily adaptable for all the researchers in the field of protein structural biology. In particular, the relative importance of each residue in the form of conservation scores (as retrieved from ConSurf [5]) and mutation/natural variant information (as retrieved from UniProt [6]) are highlighted and displayed on the structure. This way, the user can directly compare network-based residue metrics with other types of biological information. Scatter plot matrices are also available, offering the comparison of all metrics numerically in one plot, which are downloadable as well.

ProSNEx tool is open to all users without login requirement at *<<http://www.prosnex-tool.com>>*.

2.1. Materials

The tool is implemented in JavaScript and jQuery, the dynamic programming language of Hypertext Markup Language (HTML) and the modern web. The web-interface is built using Bootstrap CSS style. Network nodes and edges are constructed by JavaScript. In order to visualize the network, Cytoscape.js [94] framework is used. 3D structure of proteins is shown by PV [123] which is a WebGL-based protein viewer, and the default protein viewer of SWISS-MODEL [124].

The web-tool imports experimentally analysed or computationally modelled protein structure in PDB format as a starting point for network analysis. It supports single and multi-chain proteins or protein complexes. The calculations to construct the network are done using JavaScript, jQuery and Cytoscape.js framework. The input parameters and the results are presented by using frontend HTML pages with Bootstrap CSS library. Network analyses are conducted by JSNetworkX [120] library.

2.2. Methods

In this subsection, each step of this tool is explained in detail.

2.2.1. Data input

The first stage to construct network is to retrieve the protein structure data. The user uploads a PDB file from their local machines by either typing the 4 letter code of the PDB or by uploading a custom PDB file. The tool developed under this thesis study supports the output of a structure obtained from any modelling application in PDB file format, or the files supplied from the Protein Data Bank [1]. To analyse the 3D structure, vertices are defined with their cartesian coordinates of the representative atoms for each residue in the selected type of network, using the given details in any PDB file. Both single and multichain molecules are supported. The chain(s) of interest can be selected by the user. Depending on the type of analysis to be performed, the user can select network type (Carbon-alpha Carbon-beta) and a distance threshold for network construction. Weighted networks can also be constructed by using either the residue-residue interaction energies

in the format returned by gRINN [2] or the dynamical cross correlations from Normal Mode Analysis (NMA) [3] of the protein structure which are automatically retrieved from the PDBj database [4].

Once a network is constructed, various analyses are performed by selecting related functions from the menu bar.

2.2.2. Identifying the network parameters

The global properties representing the topology of the network such as the number of nodes, edges, average degree, density, clustering coefficient and average path lengths are displayed as the preliminary results of the network analysis.

2.2.3. Network construction and visualization

Visualization is an important aspect in the protein structure–function subject. This web tool provides mainly two windows for the visualization of the network; 3D protein structure and residue interaction network. For the latter, protein residues are represented as nodes in the graph. Adjacent nodes are connected by weighted or equally important (unweighted) edges. When a user selects a node on the network view, the corresponding residues on 3D protein structure are also highlighted.

In particular, the relative importance of each residue in the form of conservation scores (as retrieved from ConSurf [5]) and mutation/natural variant information (as retrieved from UniProt [6]) can also be highlighted and displayed on the structure.

The 3D protein structure is displayed in a ribbon diagram format by default using PV [123] WebGL 3D molecular viewer library. When user runs a command to evaluate the shortest path between two nodes, the optimal path with related nodes is automatically demonstrated in different colors based on the highlight mode.

2.2.4. Centrality analysis

Centrality analysis is one of the most important components in this study. Users can investigate the degree centrality, closeness centrality, betweenness centrality analysis within this tool. In order to perform this centrality analyzes, JSNetworkX [120] library is

used, which is an extension of NetworkX [121]. The values are plotted, displayed on the structure and are also available for download.

2.2.5. Shortest path analysis

Shortest paths between selected two nodes can be obtained using Dijkstra [40], Floyd–Warshall [39], Bellman–Ford [41] algorithms. The user can choose the calculation algorithm from a drop down menu. After the shortest paths are evaluated, they are visualized both on the interaction energy network graph and on the 3D structure of the protein or protein complex.

2.2.5.1. Dijkstra's shortest path algorithm

Dijkstra's [40] algorithm for shortest path analysis is used to detect the minimum length for a given residue to all other residues by eliminating out the lowest one in both directed and undirected and connected graph of interaction with non-negative edges weights. The algorithm stops when it finds the shortest distance between any given source and target residues. The output of Dijkstra's shortest path algorithm is a value of the total distance between any given nodes. Dijkstra's shortest path algorithm runs in with $\theta(V^2)$ time complexity. Figure A.1 is a pseudocode of this algorithm.

2.2.5.2. Floyd–Warshall algorithm

Floyd-Warshall [39] algorithm is an algorithm to detect all-pairs shortest paths in a negative or positive edge weighted graph. Once the algorithm is started to run, all the shortest paths of all pairs of vertices are calculated. The algorithm returns the total distance between the source and target nodes. The algorithm has $\theta(V^3)$ time complexity. Figure A.2 is a pseudocode of this algorithm.

2.2.5.3. Bellman–Ford algorithm

Bellman-Ford [41] algorithm is closely related to Dijkstra's shortest path algorithm [40]; however, this algorithm works for graph with negative edge weights. This algorithm can be applied for both weighted and unweighted graphs. Bellman-Ford algorithm can detect existing negative edge cycles in the graph. The computational time complexity is $\theta(VE)$,

where V is the number of vertices and E is the number of edges in the graph. Figure A.3 is a pseudocode of this algorithm.

2.3. Scatter Plot, Contact Map, Distance Map

Scatter plot [125] refers to the relationship which is called correlation between two metrics. This visualization method eases to make comparison in specific variables in analyzing structural data. Each part of a scatter plot is constructed from two numeric variable. If the correlation of these metrics reveals a straight line starts from the origin to high x and y values, in this case this is said to be a positive correlation between the variables. If the line continues from a high y -value down to a high x value, then the variables are said to have a negative correlation in this time. Figure 2.1 is an example scatter plot shows the correlations among the degree centrality, betweenness centrality, closeness centrality, clustering coefficient, and ConSurf [5] scores generated with an unweighted network for ubiquitin protein deposited at PDB [1] with 1UBQ [126] code.

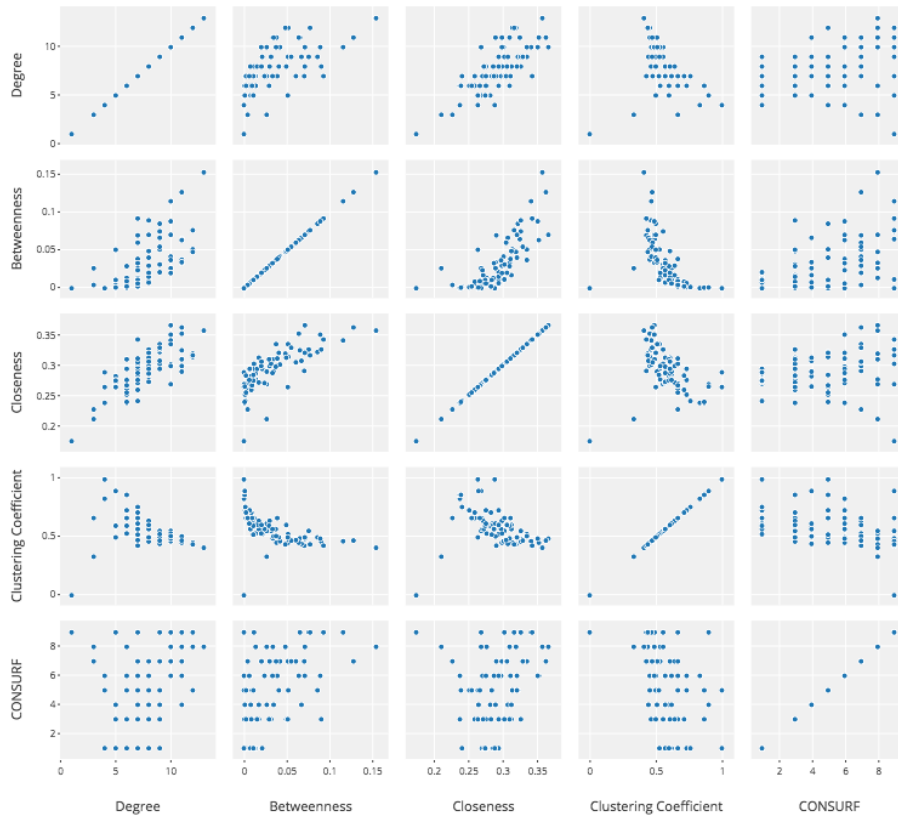


Figure 2.1 A sample scatter plot.

A *protein contact map* stands for the distance between all amino acid residue pairs in a protein structure. This is a reduced visualization type of the structure rather than showing all 3D atomic coordinates. This is also used for describing the similarity between the structures [127]. For a residue pair with a node i and j , if i and j are closer to each other in a specified threshold, ij^{th} element of the matrix is 1, and 0 otherwise. Figure 2.2 is an example contact map for the network previously created in Figure 2.1.

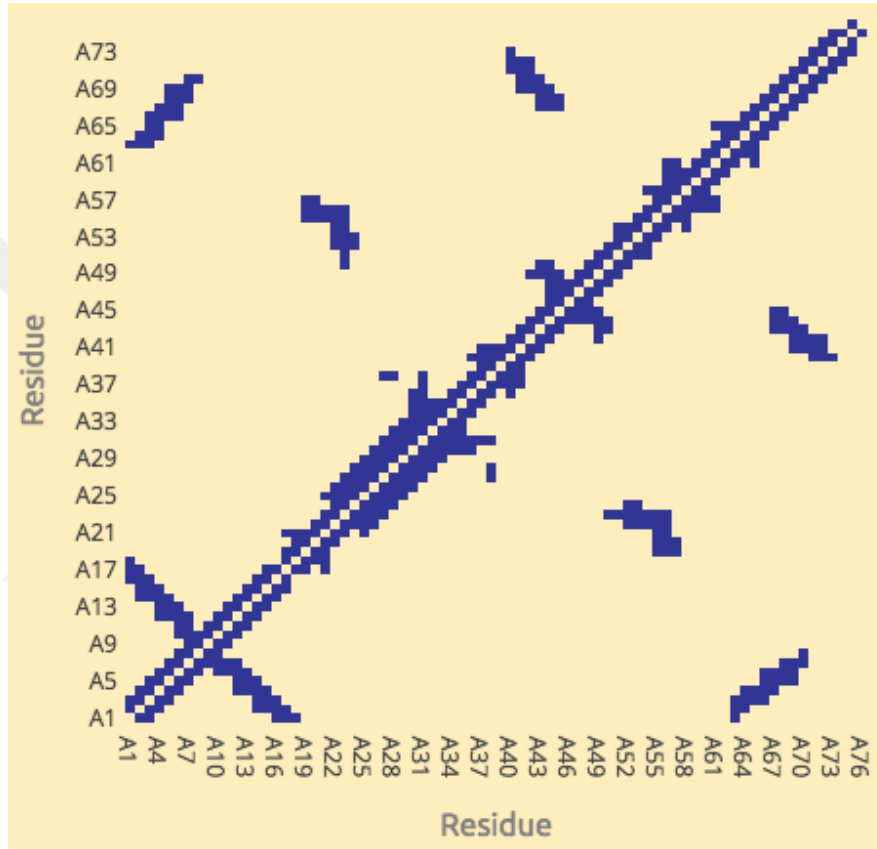


Figure 2.2 Contact map demonstration.

In a *distance map*, the distances between all the residue pairs are visualized with a gradient colors. This is said to be a generalization of a protein contact map. Figure 2.3 represents an example for distance map of ubiquitin protein residue interaction network created in Figure 2.1.

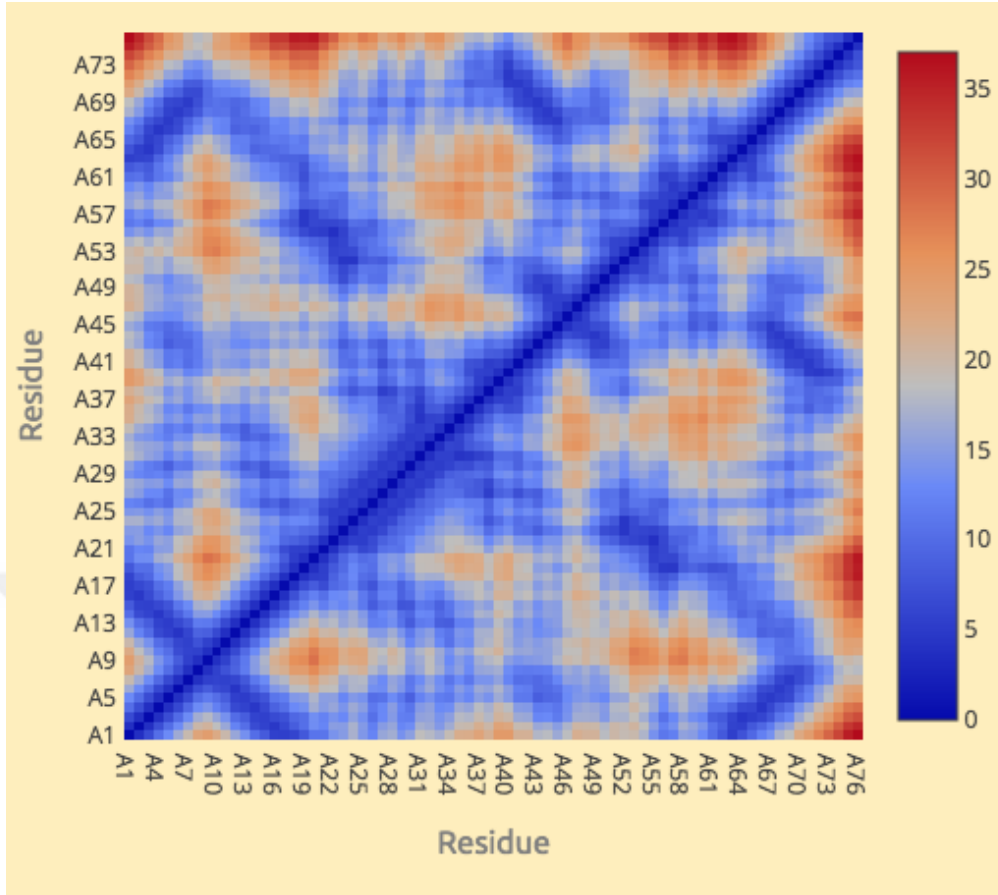


Figure 2.3 Representation of a distance map.

2.4. k -clique Concept

A *clique* [128]–[130] stands for the subsets of k nodes which k nodes are completely jointed in a graph. This reveals all the adjacent k vertices which are also called complete subgraphs.

2.5. Edge Weight in ProSNEx

In a weighted network, edges of a graph have a numerical value which are called as the edge weight. ProSNEx offers the users to define several data as edge weights, which are the residue-residue interaction energies obtained from gRINN [2] or the dynamical cross correlation data from Normal Mode Analysis (NMA) [3] of the protein structure which will be automatically retrieved from the PDBj [4] database. This is a novel and unique feature in protein structure network analysis.

This study represents a flexible and an efficient tool for assessing the significance of network properties through a biologically meaningful data. Following this aim, we have developed a web-based application tool called ProSNex (Protein Structure Network Explorer) which provides a novel and enhanced analysis of protein structures using network formalism. It provides a wide range of data and visualization options. Thus, we believe this tool is useful for the initial analysis of reconstructed networks, as well as subsequent network-based research.

The general usage of the tool is explained in detail in this section. Screenshots of the windows are also provided to guide the users.

3.1. Select PDB File

Network construction is based on a PDB file that is either provided with a PDB ID or alternatively can be uploaded by the user.

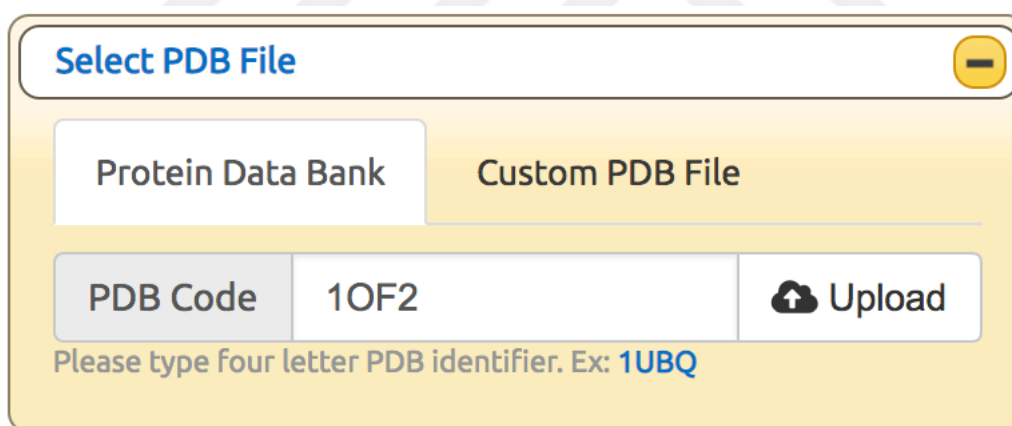


Figure 3.1 Select PDB file window.

Once a PDB is provided by the user, a chain selection window will appear for multi-chain structures. An example is provided in Figure 3.2 for PDB ID 1OF2 [131].

Select chain to analyse

	Chain	Molecule	Organism	Gene
☒	A	HUMAN LYMPHOCYTE ANTIGEN HLA-B27		
☒	B	BETA-2-MICROGLOBULIN		
☐	C	VASOACTIVE INTESTINAL POLYPEPTIDE RECEPTOR		
☐	All			

Done

Figure 3.2 Chain selection window.

3.2. Network Construction Settings

Network Settings

Network

Network Type

Carbon-alpha

Lower
0
7
Upper

Carbon-beta

Lower
0
7
Upper

Weight Type

Use interaction energies from gRINN? to construct a PEN

Choose file
No file chosen

Use NMA cross correlations from PDBj as edge weights

Unweighted network

Create

Figure 3.3 Network settings. There are various network construction options provided for the users.

There are various network construction options provided for the users.

- **Carbon-alpha**

C-alpha atoms of each amino acid are assigned as nodes. Edges are constructed between a pair of C-alpha atoms based on the upper (7 Å) and lower (0 Å) thresholds chosen by the user.

- **Carbon-beta**

C-beta atoms of each amino acid are assigned as nodes. Edges are constructed between a pair of C-beta atoms based on the upper (7 Å) and lower (0 Å) thresholds chosen by the user.

- **Edge weights from gRINN**

The tool provides the option of using Interaction Energy Matrices (IEM) given by gRINN [2] to construct a network.

- **Use NMA cross correlations from PDBj as edge weights**

This option provides the usage of Normal Mode Analysis (NMA) [3] information found in PDBj [4] database for the construction of a network.

- **Unweighted Network**

In this option, all edges are considered equally important.

3.3. Main Screen: 3D Structure of the Molecule, Residue Interaction Network and Analysis Tools

Once the network creation method is selected, three main window panels will appear on the screen.

3.3.1. 3D structure of the molecule

In this panel, 3D structure of the protein can be visualized. On hovering the mouse over a residue, the residue is colored in red and is also highlighted on the network.

3.3.2. Residue interaction network

In this panel, the constructed network can be visualized. When a node is selected by clicking, the corresponding residue also gets highlighted on the 3D structure of the protein.

3.3.3. Analysis tools

This is the main window where the user can reach all the network analysis results and visualization options of these results. A detailed explanation of this window and contents will be provided below.

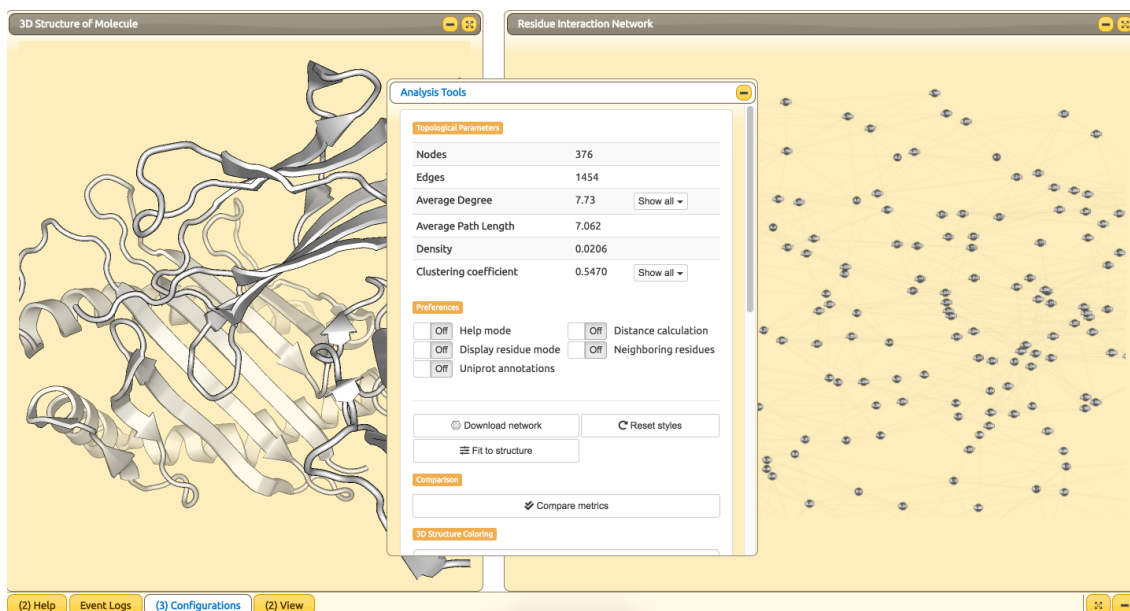


Figure 3.4 Three main window panels of ProSNEx.

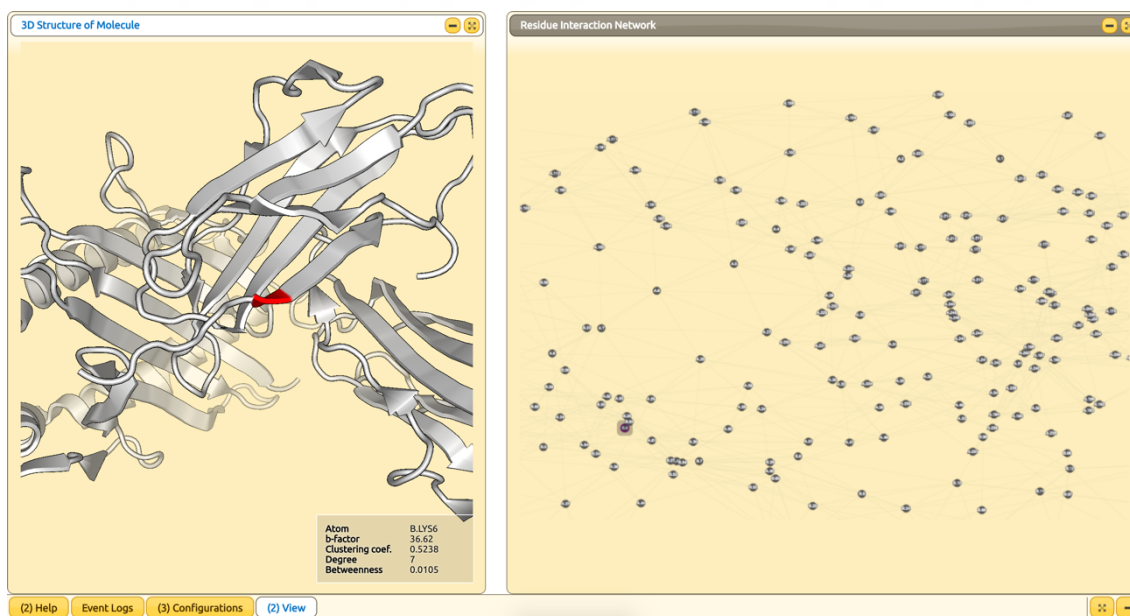


Figure 3.5 3D structure of the molecule.

There are 4 tabs at the bottom of this window.

3.3.4. About and help panels

This is the section where the user can reach the citation of the tool. It also contains a HELP section for the users.

3.3.5. Event logs

The user can follow the log from this panel.

3.3.6. Configurations panel

It will enable the user to navigate between Select PDB File, Network Settings and Analysis Tools panels.

3.3.7. View

This panel will aid the user to navigate between the 3D structure of the molecule and the Residue Interaction Network windows.

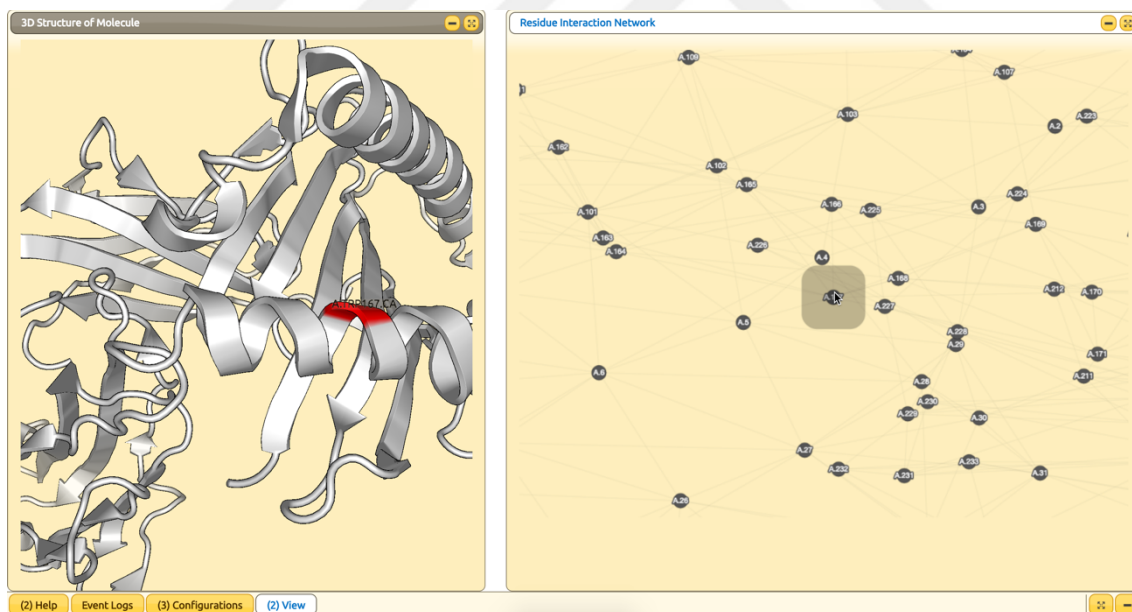


Figure 3.6 Residue interaction network.

3.3.8. Analysis tools panel

The details of the “Analysis Tools Panel” is provided below.

3.3.8.1. Topological parameters

Topological Parameters		
Nodes	376	
Edges	1454	
Average Degree	7.73	Show all ▼
Average Path Length	7.062	
Density	0.0206	
Clustering coefficient	0.5470	Show all ▼

Figure 3.7 Topological parameters.

Nodes

It is the number of nodes in the constructed network, which is equivalent to the number of residues in the structure.

Edges

The number of edges in the network.

Average degree

It is the average degree of all the nodes in the network. Individual degrees of all residues are also available. The values can be chosen to be sorted by residue or value, where *.txt* download option of the values is also available. Alternatively the hub residues can be shown on the structure or the values can be plotted to be examined in more detail.

Topological Parameters		
Nodes	376	
Edges	1454	
Average Degree	7.73	Show all ▼
Average Path Length	7.062	↕ Sort by residue
Density	0.0206	↑↓ Sort by value
Clustering coefficient	0.5470	🔗 Show hubs
		📊 Plot

Figure 3.8 Average degree tab and the drop-down menu of this property.

Average path length

It is the average of the shortest paths between all node pairs in the network. It is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. It is a measure of efficiency of information or mass transport on a network.

Density

Density of the network is defined as a ratio of the number of edges (E) to the number of possible edges in a network within N nodes.

Clustering coefficient

Clustering coefficient of the network is a critical metric for describing the characteristic structure of a protein. Individual clustering coefficients of all residues are also available. The values can be chosen to be sorted by residue or value, where *.xlsx* download option of the values is also available. The values can also be plotted to be examined in more detail.

Display residue mode

When this mode is 'ON', residue properties (residue number and type, b-factor, eccentricity, clustering coefficient, degree and betweenness properties) are shown on the 3D structure panel and the chosen residue is also highlighted on the network.

Clustering Coefficient

Clustering coefficient of the network is a critical metric for describing the characteristic structure of a protein. Individual clustering coefficients of all residues are also available. The values can be chosen to be sorted by residue or value, where [xlsx](#). download option of the values is also available. The values can also be plotted to be examined in more detail.

Clustering coefficient 0.5470 [Show all ▼](#)

Preferences

☒ On	Help mode	☐ Off	Distance calculation
☐ Off	Display residue mode	☐ Off	Neighboring residues
☐ Off	Uniprot annotations		

[Download network](#) [Reset styles](#)

[Fit to structure](#)

Comparison

☒ [Compare metrics](#)

3D Structure Coloring

Figure 3.11 An example of the help mode.

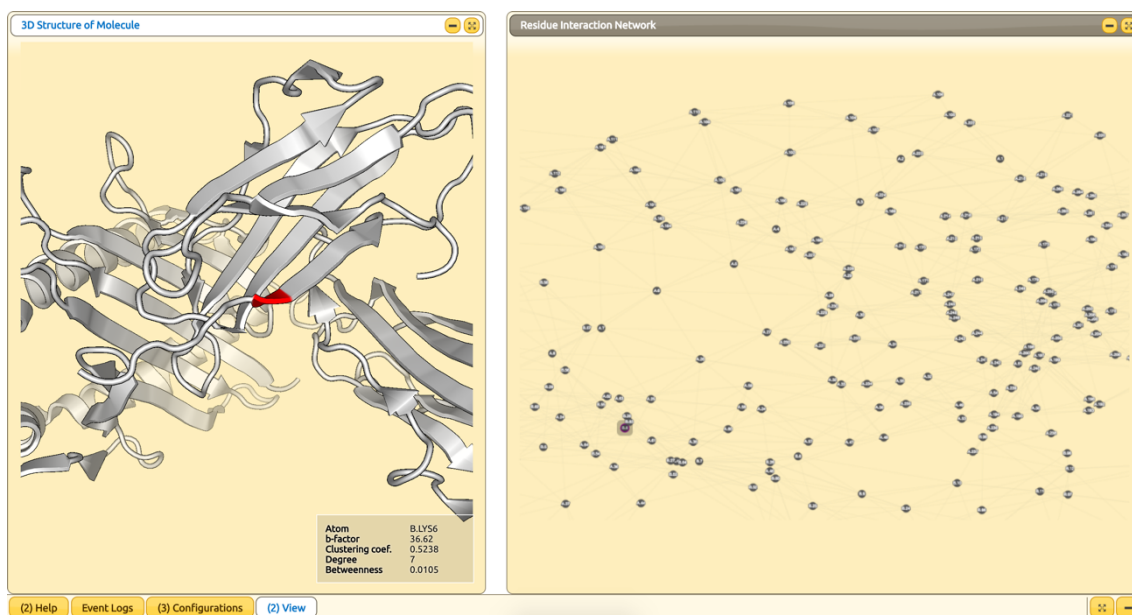


Figure 3.12 Display residue mode.

Distance calculations

When this mode is 'ON', distance between any two selected residues is calculated and shown on the structure. The chosen residues are also highlighted in the network view.

Neighboring residues

When this mode is 'ON', the neighboring residues of a selected node can easily be identified. The neighbors are highlighted in the network view and are also shown on the 3D structure. The node is shown in red and the neighboring residues are shown in purple.

UniProt annotations

This unique feature displays the natural variant and disease mutations data taken from UniProt [6] for the given PDB. The annotations are shown on the 3D protein structure and the details are listed on the screen.

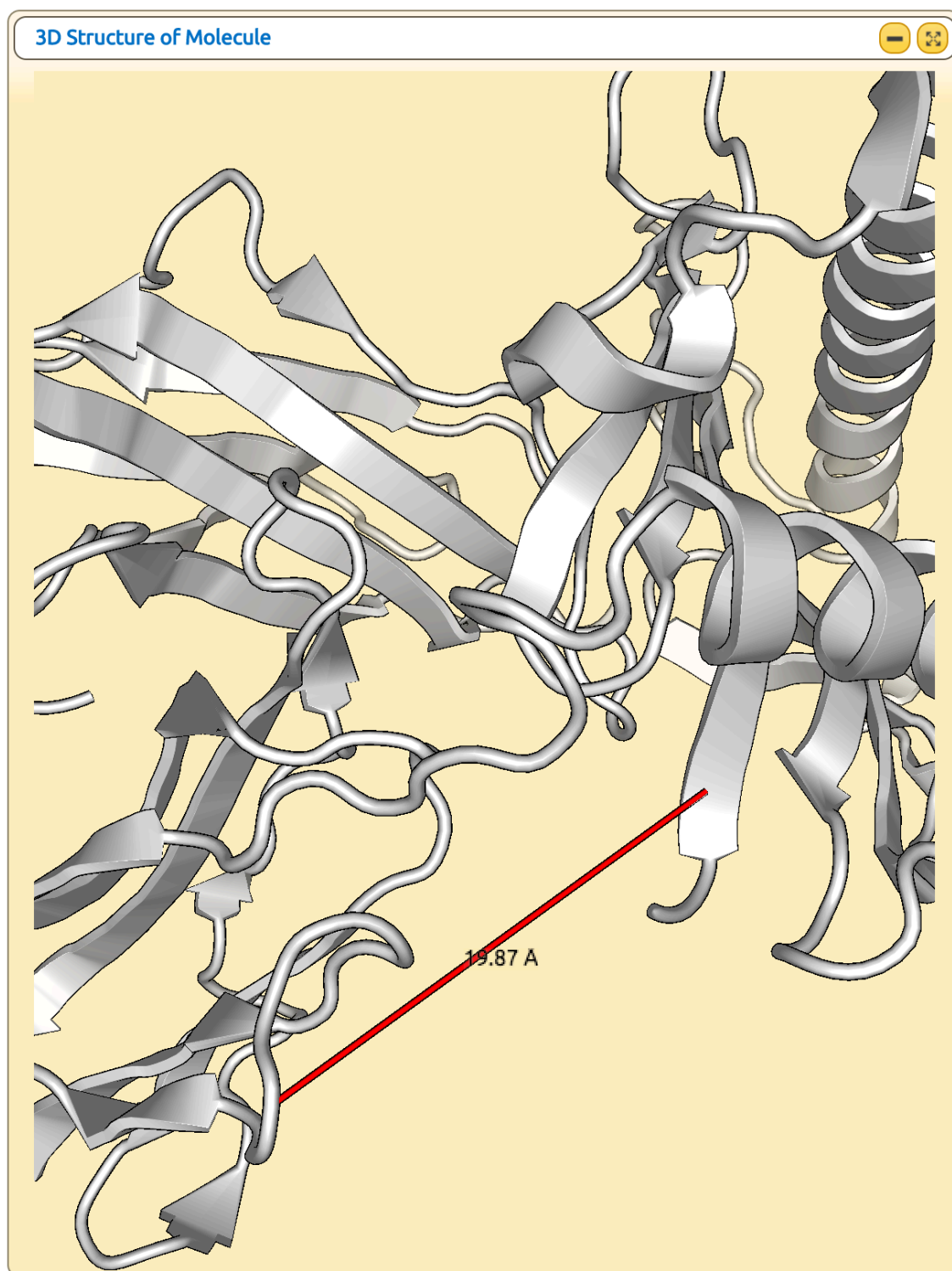


Figure 3.13 Distance calculation between residues A.PRO267 and A.HIS3 for 1OF2 [131].

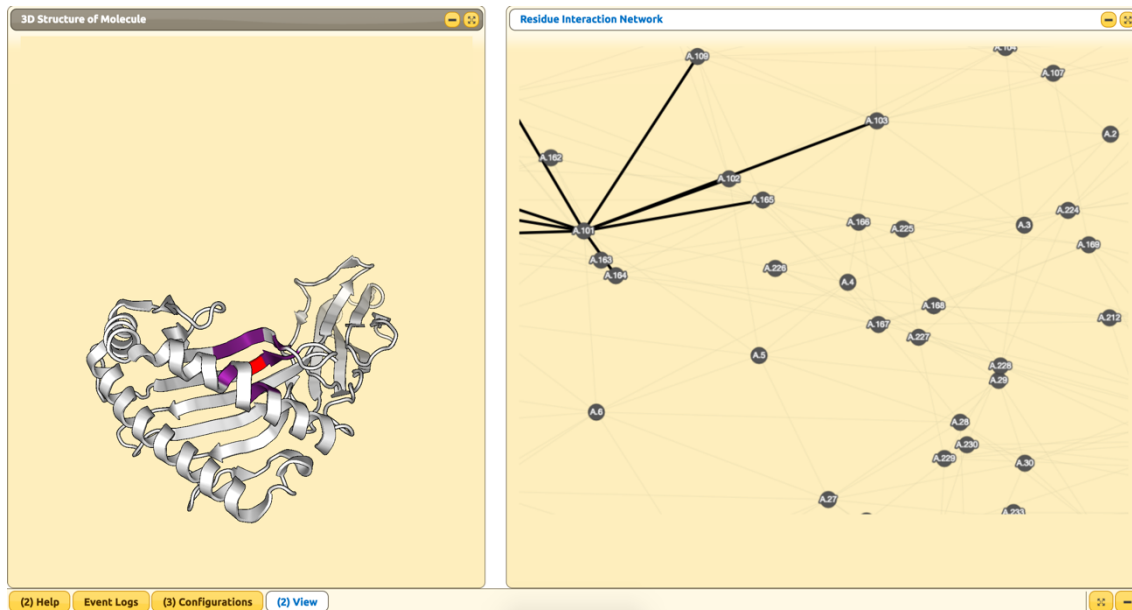


Figure 3.14 Display of the neighboring residues mode. Chain A, residue 101 is selected from the network panel, the corresponding neighbors are displayed on the 3D structure.

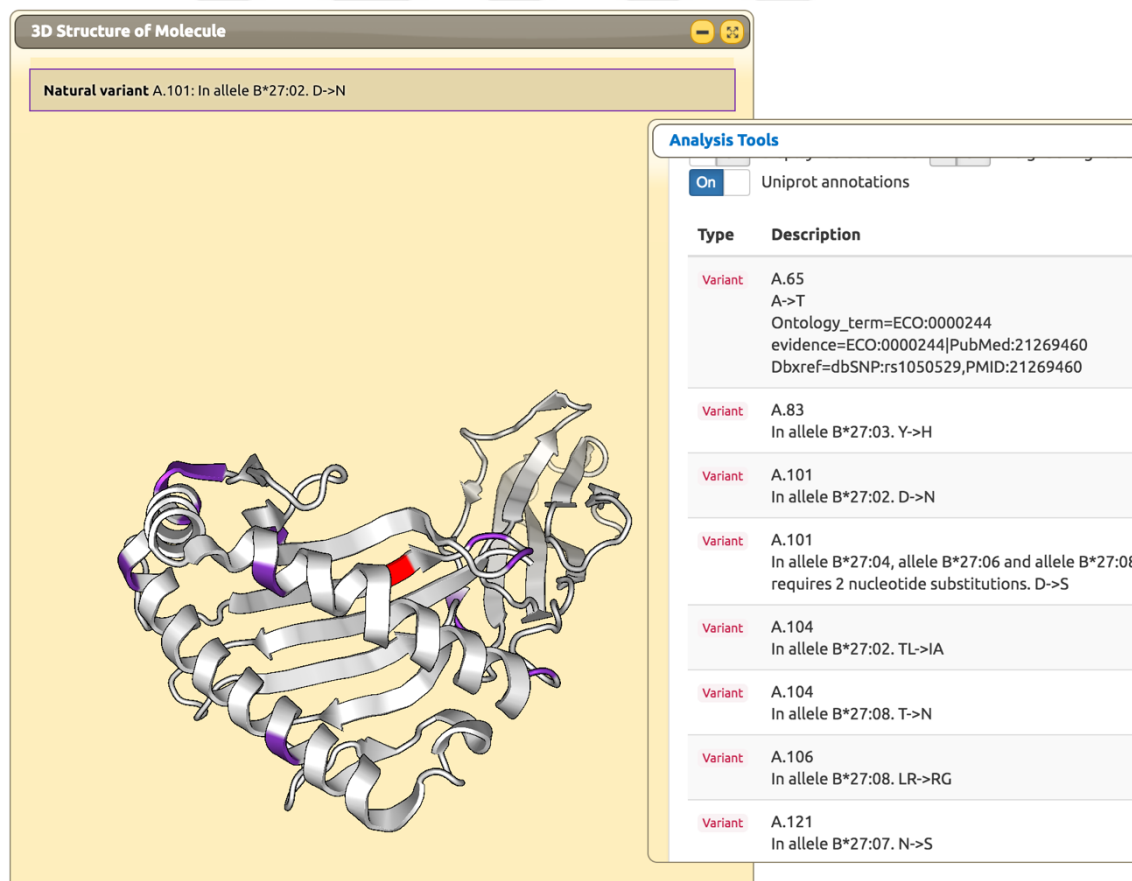


Figure 3.15 UniProt annotations are marked on the 3D structure and are also listed.

3.3.8.3. Comparison

Comparison section enables the users to compare and visualize certain metrics (b-factor, conservation data, degree, closeness centrality and betweenness centrality) on the structures. When 'compare metrics' is clicked, a new window appears. There are two drop-down menus available for the user. 3D structure of the protein is colored according to the choice of the user.

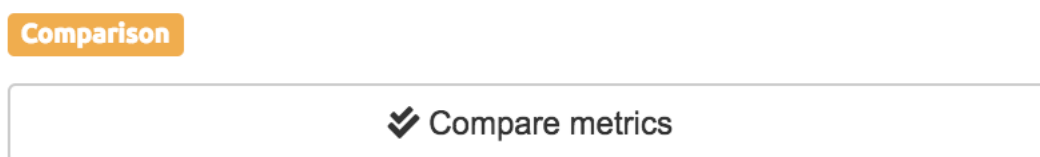


Figure 3.16 Comparison section, compare metrics button.

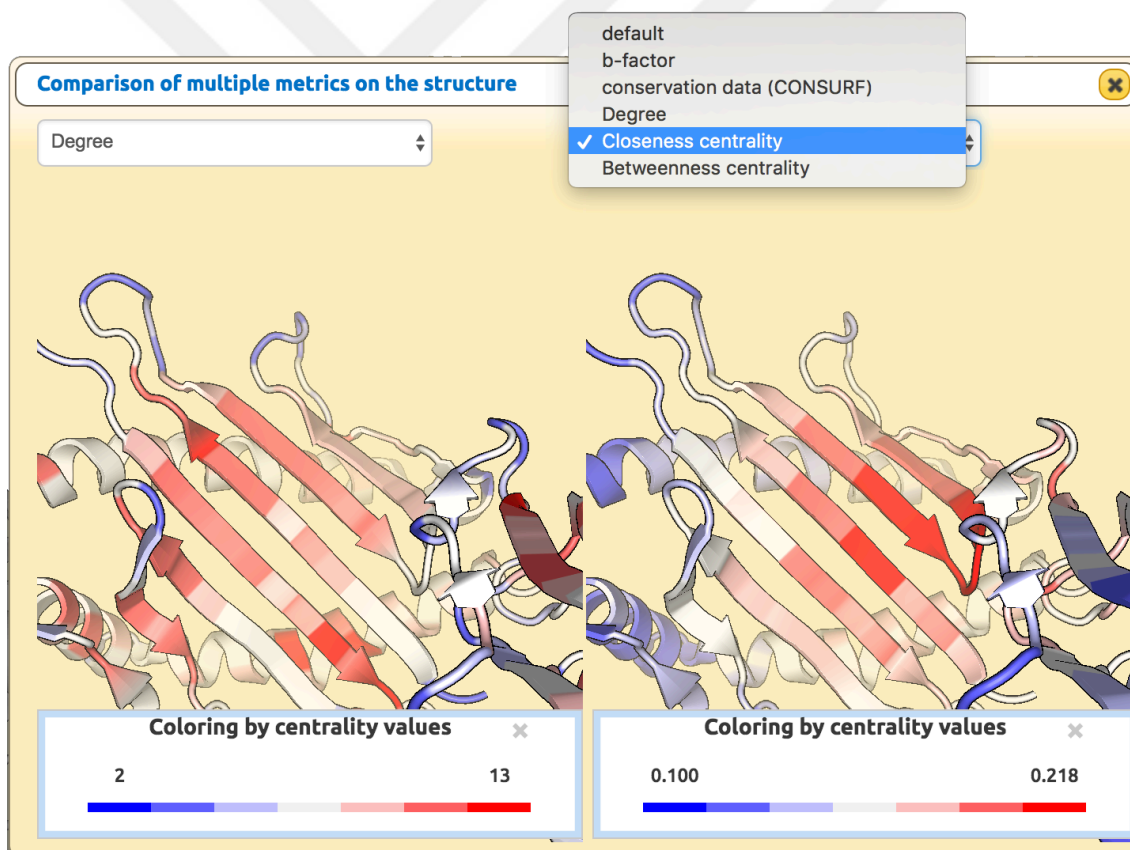


Figure 3.17 Comparison of multiple metrics on the same structure can be done using the 'compare metrics' button.

3.3.8.4. 3D structure coloring

Structure coloring based on b-factor (taken from PDB file), conservation scores (taken from ConSurf [5]) and network metrics (degree, closeness centrality and betweenness centrality) is possible.

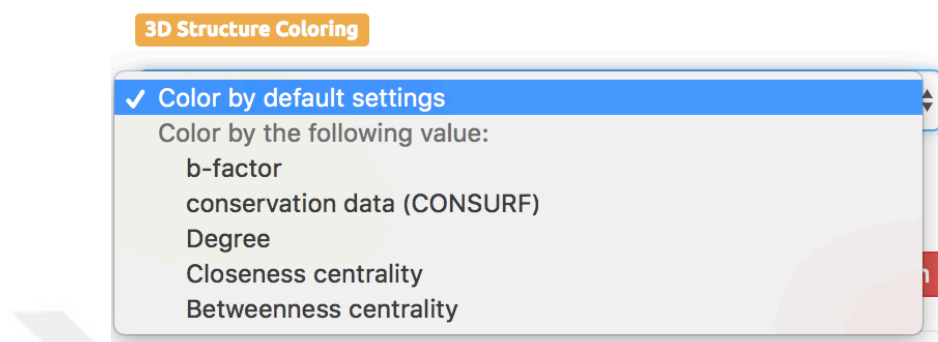


Figure 3.18 Drop-down menu for the 3D structure coloring section.

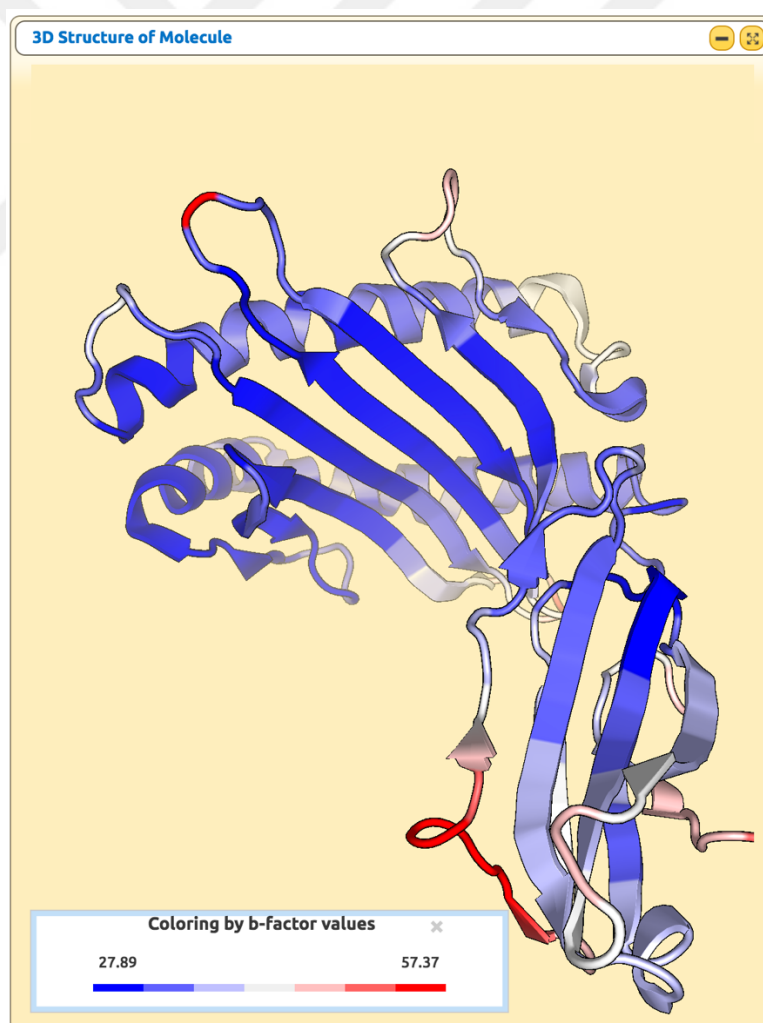


Figure 3.19 Coloring of the 3D structure based on b-factor values for 1OF2/A.

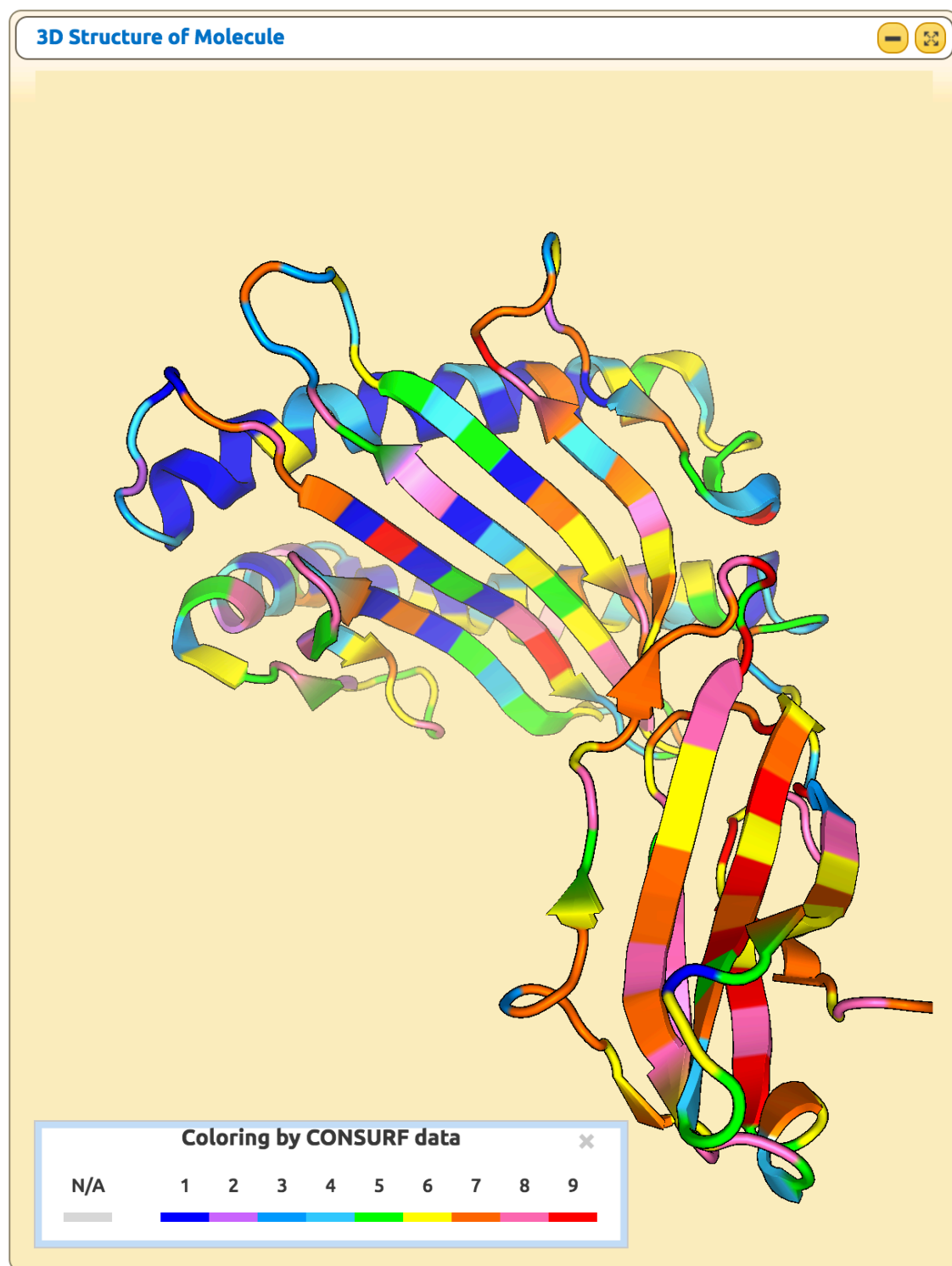


Figure 3.20 Coloring of the 3D structure based on conservation values taken from ConSurf for 1OF2/A.



Figure 3.21 Coloring of the 3D structure based on centrality values for 1OF2/A.

3.3.8.5. Additional features

In the preferences section, there are some additional features.



Figure 3.22 Some additional features of the preferences section.

Download network

This button enables the users to download the network constructed by ProSNEx in a *.json* format.

Reset styles

Resets all the styles present in the network and 3D structure of the molecule panels.

Fit to structure

Zooms out the 3D structure and resets the view.

3.3.8.6. Shortest path algorithm

Shortest path between two vertices is the path that has the minimum sum of edge weights. The user can display the shortest path(s) by choosing the residues of interest from the drop-down menu. Additionally, ProSNEx offers three different shortest path algorithms (Dijkstra [40], Floyd-Warshall [39] and Bellman-Ford [41]) for the users, which can also be chosen from the drop down menu.

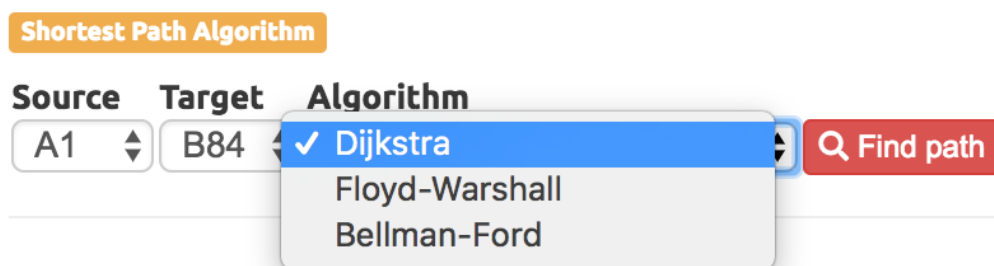


Figure 3.23 Shortest path algorithm drop down menu.

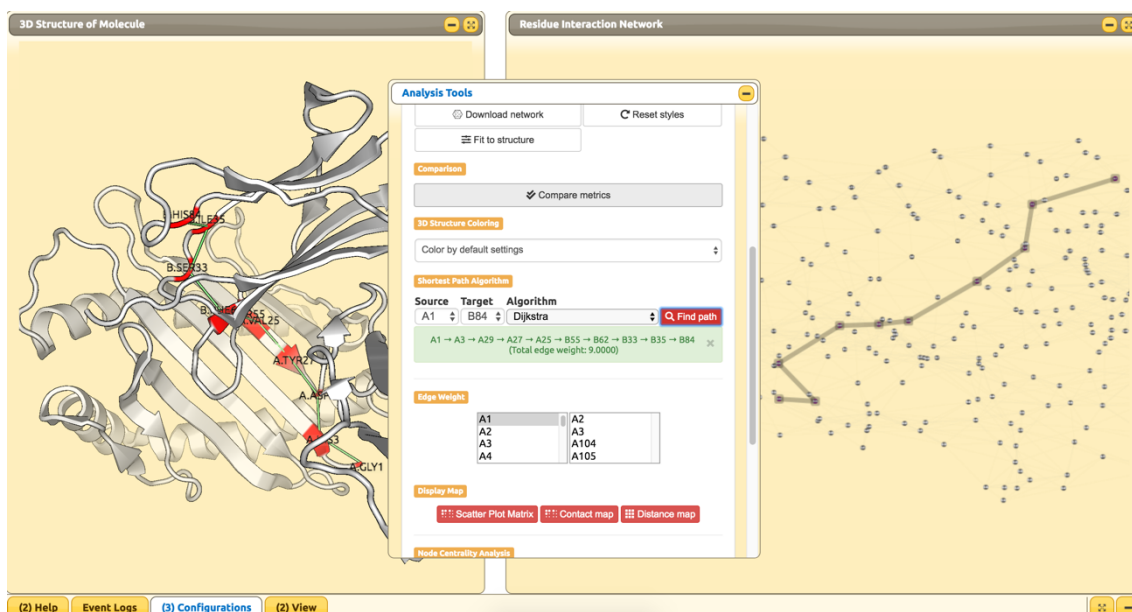


Figure 3.24 An example shortest path analysis result is displayed for 1OF2, between residues A1 and B84. The total edge weight is given and the path is displayed on both the 3D structure and the network view panels.

3.3.8.7. Edge weight

For the chosen network construction type, edge weights are calculated accordingly. For any two chosen residues, value of the edge weight is displayed and these residues are highlighted on the network view panel.

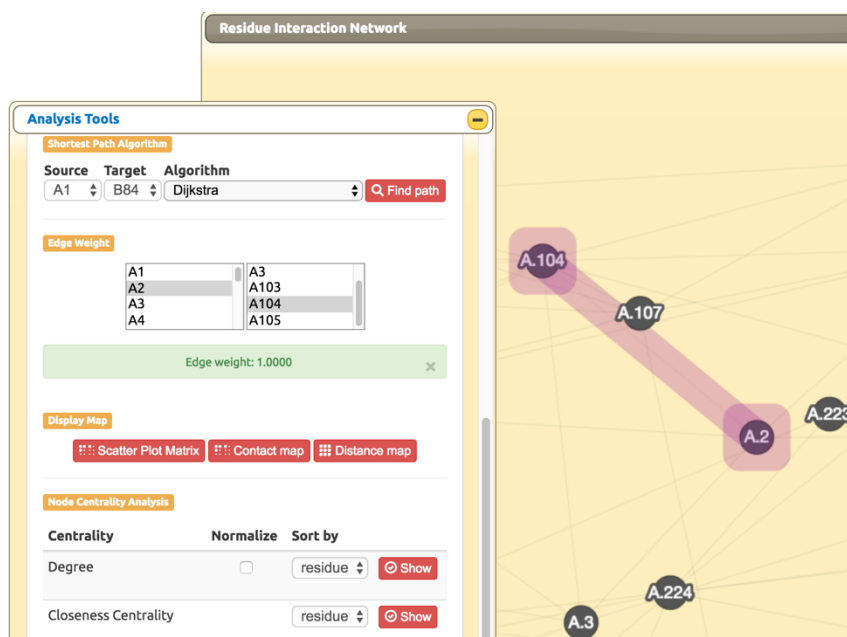


Figure 3.25 Display of edge weight option.

3.3.8.8. Display map

In this section, there are three maps that can be displayed.



Figure 3.26 Display map section.

Scatter plot matrix

This highly informative scatter plot matrix displays scatter plots of all the network metrics and parameters (degree, betweenness, closeness, clustering coefficient and ConSurf scores) against each other. Data regarding the value of each can also be downloaded in a *.txt* and *.x/sx* formats.



Figure 3.27 Display of the scatter plot matrix for 1OF2 [131].

Contact map

Contact map is a display of the network in a 2D matrix view. Each dot represents an edge between the i^{th} and j^{th} residues. 2D representation of the matrix is available for download in both *.txt* data format and *.png* formats. When a cell is chosen in the contact map, it is also highlighted on the 3D structure and the network view in order to ease the analysis for the user.

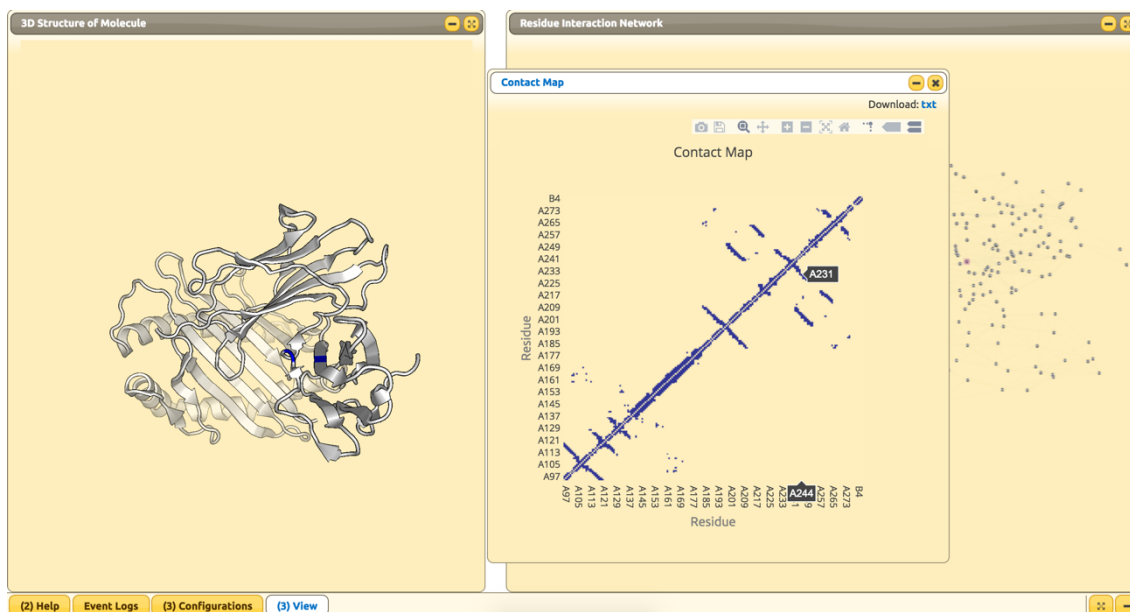


Figure 3.28 Display of the contact map for 1OF2 [131].

Distance map

Displays the distance map between the amino acid residues. It is a matrix of elements representing the distances between any two residues. 2D representation of the matrix is available for download in both *.txt* data format and *.png* formats.

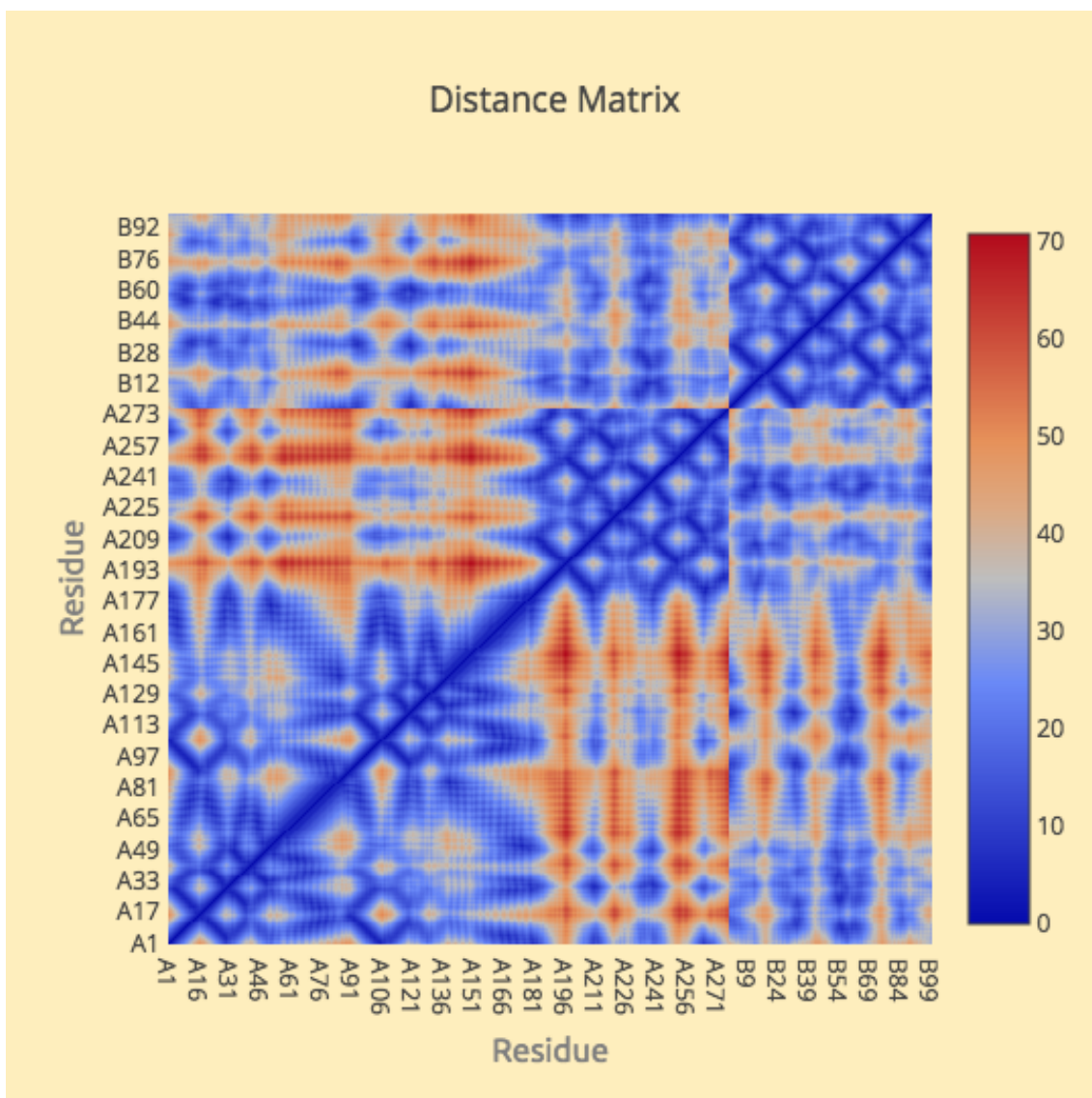


Figure 3.29 Display of the distance map for 1OF2 [131].

3.3.8.9. Node centrality analysis

Node centrality analysis section includes the results of degree, closeness centrality and betweenness centrality calculations. The results can be normalized or can be visualized directly. The values can be displayed by sorting them according to the residue number of metric value.

Degree centrality

Degree is the measure of the direct neighbors of a node. For a carbon-alpha network constructed for chain A of 1OF2 [131], normalized degree values are given in Figure

3.31. The values can be listed, the list can be downloaded in *.txt* and *.x/sx* formats. When a certain residue is clicked in the list, it is also shown on the 3D structure. In the network panel, nodes are also colored and sized according to the degree values. A plot option for the degree results is also available for analysis.

Closeness centrality

Closeness centrality is the inverse of the total shortest path distance of the node to all other nodes of the network. For a carbon-alpha network constructed for chain A of 1OF2 [131], closeness centrality values are given in the following figure. The values can be listed, the list can be downloaded in *.txt* and *.x/sx* formats. When a certain residue is clicked in the list, it is also zoomed in the 3D structure. In the network panel, nodes are also colored and sized according to the degree values. A plot option is also available for further analysis.

Betweenness centrality

Betweenness centrality is the ratio of shortest paths passing through the node. For a carbon-alpha network constructed for chain A of 1OF2 [131], normalized betweenness centrality values are given in the following figure. The values can be listed, the list can be downloaded in *.txt* and *.x/sx* formats. When a certain residue is clicked in the list, it is also shown by zooming in the 3D structure. In the network panel, nodes are also colored and sized according to the degree values. A plot option is also available for further analysis.

Node Centrality Analysis

Centrality

Normalize

Sort by

Degree
☐
residue
Show

Closeness Centrality
residue
Show

Betweenness Centrality
☐
residue
Show

Figure 3.30 Node centrality analyses section.

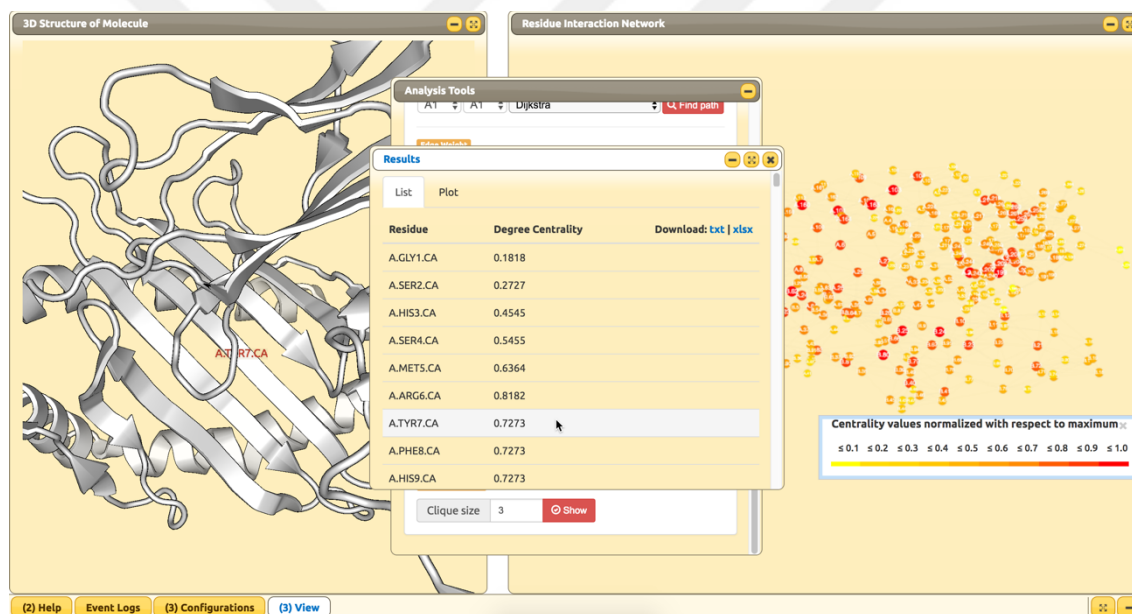


Figure 3.31 Degree centrality results of a network constructed for 1OF2/A and 1OF2/B.

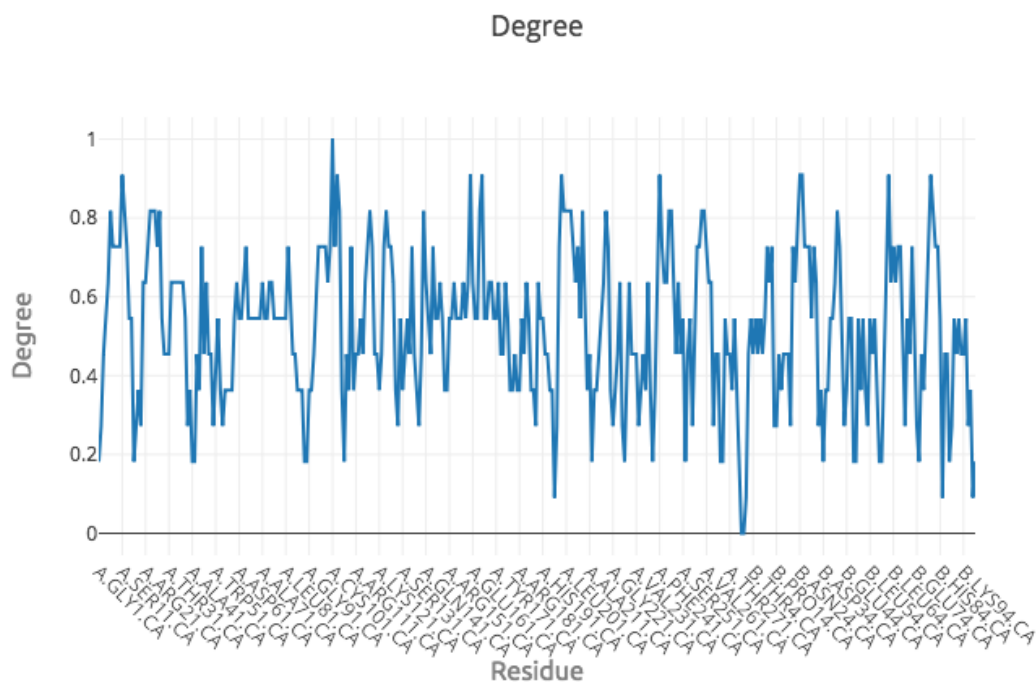


Figure 3.32 Plot of degree centrality values.

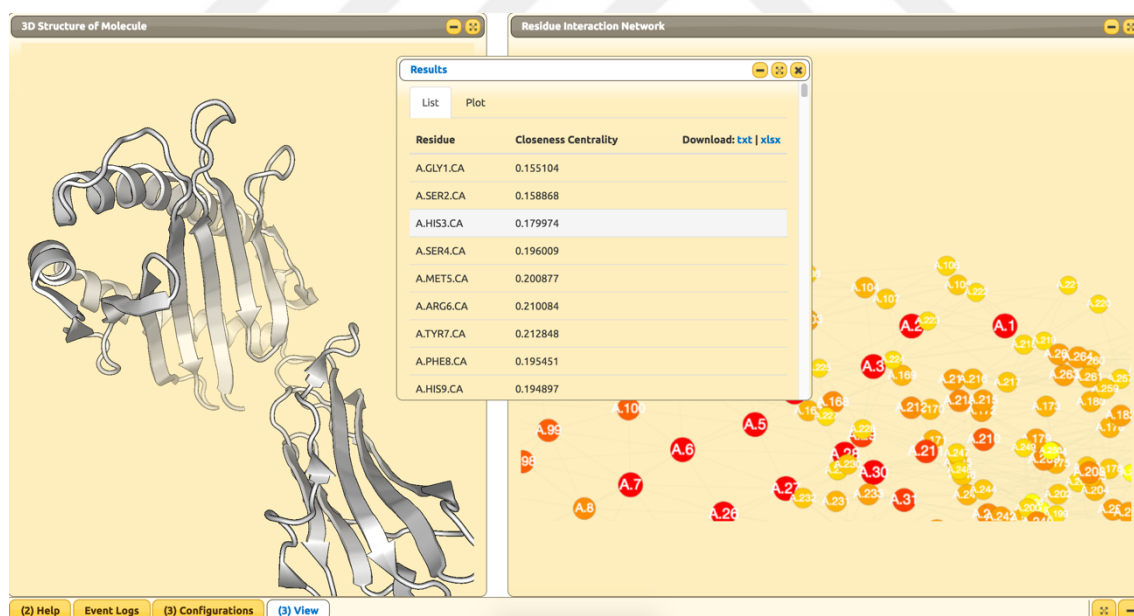


Figure 3.33 Closeness centrality results of a network constructed for chain A of 1OF2.

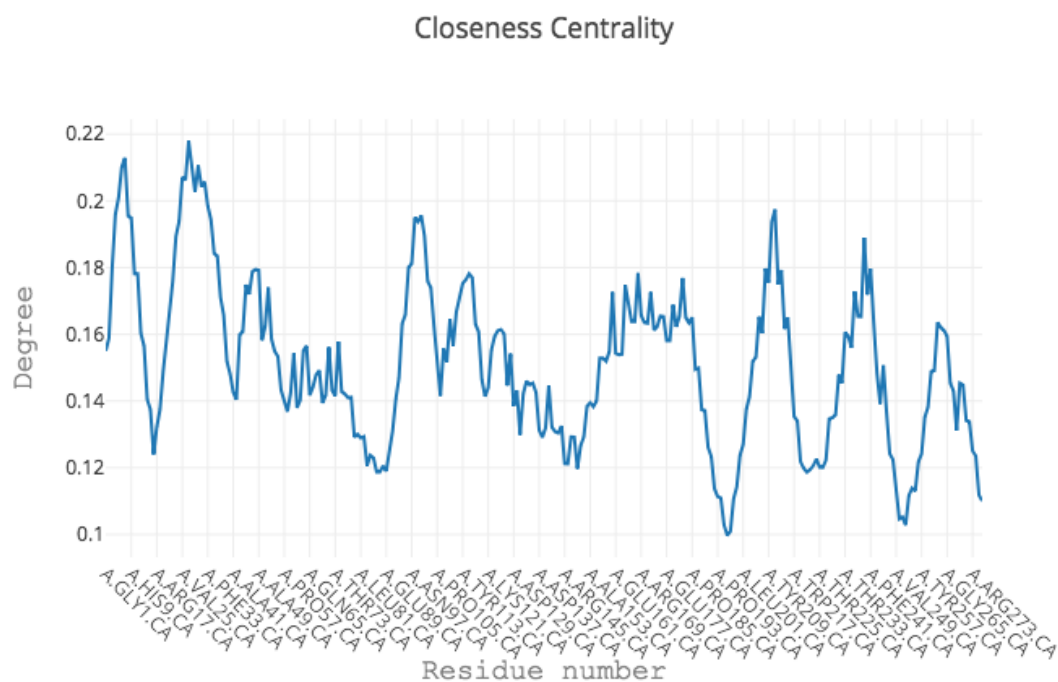


Figure 3.34 Plot of closeness centrality results for a network constructed for 1OF2/A.

3.3.8.10. k -clique analysis

k -clique defines the sub-network of k nodes with all the k nodes connected to each other. The size of the k can be entered by the user. The resulting k -cliques will be listed and the selected one will be highlighted in both the 3D protein structure and the network view.

k -clique Analysis

Clique size	6	Show
-------------	---	------

Figure 3.35 k -clique analysis.

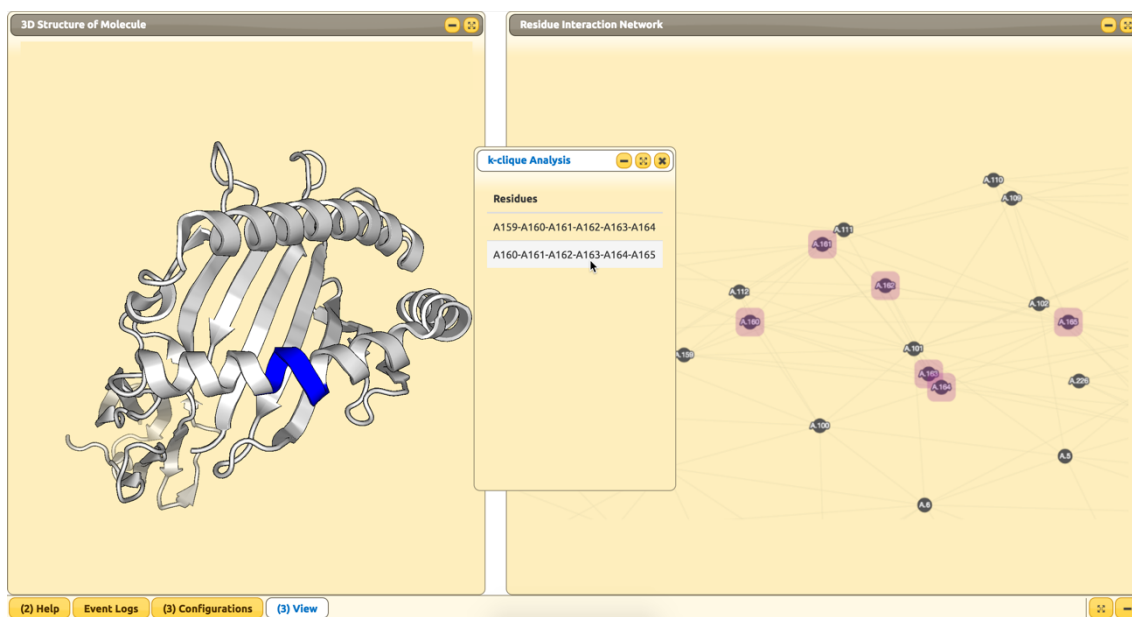


Figure 3.36 Display of k -clique analysis results for a clique size of 6 on 1OF2/A.

CHAPTER 4

CONCLUSION

Rapid rise in the development of technological inventions led structural biologist to explore protein structures in more detail with the help of accurate and immense data obtained from computational applications. Comprehensive results for structure and dynamics of proteins are getting intelligible through particular tools of network analysis. In this study, an analysis tool that works on modern web browsers is developed which only requires the protein structure file in a PDB format as an input. Upon the construction of the network based on the users' choices, the analysis results are shown on both 2D and 3D conformations at the same time. The edge weights can be modified by either providing the edge data file in interaction energy format obtained from gRINN [2] or NMA [3] values from the PDBj [4] can be used, if desired.

The tool has the ability to calculate some fundamental parameters for network topology of protein structures including betweenness, closeness, shortest path length through selecting optional shortest path algorithms (Dijkstra [40], Floyd-Warshall [39], and Bellman-Ford [41]), and k -cliques. The tool is designed to be very user friendly and easily adaptable for all the researchers in the field of protein structural biology. In particular, the relative importance of each residue in the form of conservation scores (as retrieved from ConSurf [5]) and mutation/natural variant information (as retrieved from UniProt [6]) are highlighted and displayed on the structure. This way, the user can directly compare network-based residue metrics with other types of biological information. Scatter plot matrices are also available, offering the comparison of all metrics numerically in one plot, which provide downloadable data as well.

The major novelty of the tool lies in the presence of features such as

1. Comparison of multiple single residue metrics from network analysis as well as other additional information such as ConSurf [5] conservation scores and UniProt [6] disease annotations.
2. Usage of dynamic cross-correlations between pairs of amino acids from NMA [3] as edge-weights in the constructed Protein Structure Network.

As of today, ProSNEx is the only tool that combines all these features and specifications.

This web tool can be used to highlight the crucial residues for the function of a protein structure, which is undeniably a very valuable tool for all the researchers in the field.

ProSNEx has been developed mainly in JavaScript and is available at <http://prosnex-tool.com>. It is compatible with major web browser including Chrome, Firefox and Safari.



REFERENCES

- [1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000.
- [2] O. Serçinoğlu and P. Ozbek, "gRINN: a tool for calculation of residue interaction energies and protein energy network analysis of molecular dynamics simulations," *Nucleic Acids Res.*, vol. 46, no. W1, pp. W554–W562, Jan. 2018.
- [3] H. Wako, M. Kato, and S. Endo, "ProMode: a database of normal mode analyses on protein molecules with a full-atom model," *Bioinformatics*, vol. 20, no. 13, pp. 2035–2043, Jan. 2004.
- [4] A. R. Kinjo, G.-J. Bekker, H. Suzuki, Y. Tsuchiya, T. Kawabata, Y. Ikegawa, and H. Nakamura, "Protein Data Bank Japan (PDBj): updated user interfaces, resource description framework, analysis tools for large structures," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D282–D288, Jan. 2017.
- [5] A. Armon, D. Graur, and N. Ben-Tal, "ConSurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information," *J. Mol. Biol.*, vol. 307, no. 1, pp. 447–463, Jan. 2001.
- [6] The UniProt Consortium, "UniProt: the universal protein knowledgebase," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D158–D169, Jan. 2017.
- [7] A. Vazquez, "Protein Interaction Networks," in *Neuroproteomics*, O. Alzate, Ed. Boca Raton (FL): CRC Press/Taylor & Francis, 2010.
- [8] H. J. Bernstein, "Recent changes to RasMol, recombining the variants," *Trends Biochem. Sci.*, vol. 25, no. 9, pp. 453–455, Jan. 2000.
- [9] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF Chimera—A visualization system for exploratory research and analysis," *J. Comput. Chem.*, vol. 25, no. 13, pp. 1605–1612, Jan. 2004.
- [10] D. C. Richardson and J. S. Richardson, "The kinemage: A tool for scientific communication," *Protein Sci.*, vol. 1, no. 1, pp. 3–9, Jan. 2008.
- [11] W. Humphrey, A. Dalke, and K. Schulten, "VMD: Visual molecular dynamics," *J. Mol. Graph.*, vol. 14, no. 1, pp. 33–38, Jan. 1996.
- [12] "The PyMOL Molecular Graphics System." Schrodinger, LLC.
- [13] "Jmol: an open-source Java viewer for chemical structures in 3D."

- [14] M. Vendruscolo, N. V Dokholyan, E. Paci, and M. Karplus, "Small-world view of the amino acids that play a key role in protein folding," *Phys. Rev. E*, vol. 65, no. 6, p. 61910, Jan. 2002.
- [15] L. H. Greene and V. A. Higman, "Uncovering network systems within protein structures," *J. Mol. Biol.*, vol. 334, no. 4, pp. 781–791, 2003.
- [16] G. Bagler and S. Sinha, "Network properties of protein structures," *Phys. A Stat. Mech. its Appl.*, vol. 346, no. 1, pp. 27–33, Jan. 2005.
- [17] K. V Brinda and S. Vishveshwara, "A Network Representation of Protein Structures: Implications for Protein Stability," *Biophys. J.*, vol. 89, no. 6, pp. 4159–4170, Jan. 2005.
- [18] E. Estrada, "Universality in Protein Residue Networks," *Biophys. J.*, vol. 98, no. 5, pp. 890–900, Jan. 2010.
- [19] A. R. Atilgan, P. Akan, and C. Baysal, "Small-World Communication of Residues and Significance for Protein Dynamics," *Biophys. J.*, vol. 86, no. 1, pp. 85–91, Jan. 2004.
- [20] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, no. 4–5, pp. 175–308, 2006.
- [21] N. A. Alves and A. S. Martinez, "Inferring topological features of proteins from amino acid residue networks," *Phys. A Stat. Mech. its Appl.*, vol. 375, no. 1, pp. 336–344, Jan. 2007.
- [22] P. Csermely, "Creative elements: network-based predictions of active centres in proteins and cellular and social networks," *Trends Biochem. Sci.*, vol. 33, no. 12, pp. 569–576, Jan. 2008.
- [23] C. Böde, I. A. Kovács, M. S. Szalay, R. Palotai, T. Korcsmáros, and P. Csermely, "Network analysis of protein dynamics," *FEBS Lett.*, vol. 581, no. 15, pp. 2776–2782, Jan. 2007.
- [24] D. J. Jacobs, A. J. Rader, L. A. Kuhn, and M. F. Thorpe, "Protein flexibility predictions using graph theory," *Proteins Struct. Funct. Bioinforma.*, vol. 44, no. 2, pp. 150–165, Jan. 2001.
- [25] M. Bhattacharyya, C. R. Bhat, and S. Vishveshwara, "An automated approach to network features of protein structure ensembles: An Automated Approach to Network Features of Protein Structure Ensembles," *Protein Sci.*, vol. 22, no. 10, pp. 1399–1416, Jan. 2013.
- [26] E. M. Mitchell, P. J. Artymiuk, D. W. Rice, and P. Willett, "Use of techniques derived from graph theory to compare secondary structure motifs in proteins," *J.*

- Mol. Biol.*, vol. 212, no. 1, pp. 151–166, Jan. 1990.
- [27] B. Chakrabarty and N. Parekh, “NAPS: Network analysis of protein structures,” *Nucleic Acids Res.*, vol. 44, no. W1, pp. W375–W382, 2016.
 - [28] S. H. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, no. 6825, p. 268, 2001.
 - [29] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Rev. Mod. Phys.*, vol. 74, no. 1, p. 47, 2002.
 - [30] G. Amitai, A. Shemesh, E. Sitbon, M. Shklar, D. Netanel, I. Venger, and S. Pietrokovski, “Network analysis of protein structures identifies functional residues,” *J. Mol. Biol.*, vol. 344, no. 4, pp. 1135–1146, 2004.
 - [31] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “SCOP: a structural classification of proteins database for the investigation of sequences and structures,” *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–540, 1995.
 - [32] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, “CATH—a hierarchic classification of protein domain structures,” *Structure*, vol. 5, no. 8, pp. 1093–1109, 1997.
 - [33] S. Lifson and C. Sander, “Antiparallel and parallel β -strands differ in amino acid residue preferences,” *Nature*, vol. 282, no. 5734, p. 109, 1979.
 - [34] A. A. S. T. Ribeiro and V. Ortiz, “Determination of Signaling Pathways in Proteins through Network Theory: Importance of the Topology,” *J. Chem. Theory Comput.*, vol. 10, no. 4, pp. 1762–1769, Jan. 2014.
 - [35] N. V Dokholyan, L. Li, F. Ding, and E. I. Shakhnovich, “Topological determinants of protein folding,” *Proc. Natl. Acad. Sci.*, vol. 99, no. 13, pp. 8637–8641, Jan. 2002.
 - [36] N. Kannan and S. Vishveshwara, “Identification of side-chain clusters in protein structures by a graph spectral method” Edited by J. M. Thornton,” *J. Mol. Biol.*, vol. 292, no. 2, pp. 441–464, Jan. 1999.
 - [37] R. Landgraf, I. Xenarios, and D. Eisenberg, “Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins” Edited by J. Thornton,” *J. Mol. Biol.*, vol. 307, no. 5, pp. 1487–1502, Jan. 2001.
 - [38] S. W. Lockless, “Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families,” *Science (80-.)*, vol. 286, no. 5438, pp. 295–299, Jan. 1999.
 - [39] R. W. Floyd, “Algorithm 97: Shortest path,” *Commun. ACM*, vol. 5, no. 6, p. 345, Jan. 1962.

- [40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Jan. 1959.
- [41] R. Bellman, "On a routing problem," *Q. Appl. Math.*, vol. 16, no. 1, pp. 87–90, Jan. 1958.
- [42] A. del Sol, H. Fujihashi, D. Amoros, and R. Nussinov, "Residue centrality, functionally important residues, and active site shape: Analysis of enzyme and non-enzyme families," *Protein Sci.*, vol. 15, no. 9, pp. 2120–2128, Jan. 2006.
- [43] D. Piovesan, G. Minervini, and S. C. E. Tosatto, "The RING 2.0 web server for high quality residue interaction networks," *Nucleic Acids Res.*, vol. 44, no. W1, pp. W367–W374, Jan. 2016.
- [44] L. H. Greene, "Protein structure networks," *Brief. Funct. Genomics*, vol. 11, no. 6, pp. 469–478, Jan. 2012.
- [45] M. Aftabuddin and S. Kundu, "AMINONET– a tool to construct and visualize amino acid networks, and to calculate topological parameters," *J. Appl. Crystallogr.*, vol. 43, no. 2, pp. 367–369, Jan. 2010.
- [46] M. S. Vijayabaskar, "GraProStr – Graphs of Protein Structures: A Tool for Constructing the Graphs and Generating Graph Parameters for Protein Structures," *Open Bioinforma. J.*, vol. 5, no. 1, pp. 53–58, Jan. 2011.
- [47] J.-L. Chung, J. E. Beaver, E. D. Scheeff, and P. E. Bourne, "Con-Struct Map: a comparative contact map analysis tool," *Bioinformatics*, vol. 23, no. 18, pp. 2491–2492, Jan. 2007.
- [48] D. Kozma, I. Simon, and G. E. Tusnady, "CMWeb: an interactive on-line tool for analysing residue-residue contacts and contact prediction methods," *Nucleic Acids Res.*, vol. 40, no. W1, pp. W329–W333, Jan. 2012.
- [49] A. J. M. Martin, M. Vidotto, F. Boscaroli, T. Di Domenico, I. Walsh, and S. C. E. Tosatto, "RING: networking interacting residues, evolutionary information and energetics in protein structures," *Bioinformatics*, vol. 27, no. 14, pp. 2003–2005, Jan. 2011.
- [50] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, 2003.
- [51] N. T. Doncheva, K. Klein, F. S. Domingues, and M. Albrecht, "Analyzing and visualizing residue networks of protein structures," *Trends Biochem. Sci.*, vol. 36, no. 4, pp. 179–182, Jan. 2011.
- [52] S. Nepomnyachiy, N. Ben-Tal, and R. Kolodny, "CyToStruct: Augmenting the

Network Visualization of Cytoscape with the Power of Molecular Viewers,” *Structure*, vol. 23, no. 5, pp. 941–948, Jan. 2015.

- [53] M. Pasi, M. Tiberti, A. Arrigoni, and E. Papaleo, “xPyder: A PyMOL Plugin To Analyze Coupled Residues and Their Networks in Protein Structures,” *J. Chem. Inf. Model.*, vol. 52, no. 7, pp. 1865–1874, Jan. 2012.
- [54] M. Tiberti, G. Invernizzi, M. Lambrughi, Y. Inbar, G. Schreiber, and E. Papaleo, “PyIntergraph: A Framework for the Analysis of Interaction Networks in Structural Ensembles of Proteins,” *J. Chem. Inf. Model.*, vol. 54, no. 5, pp. 1537–1551, Jan. 2014.
- [55] V. Sobolev, A. Sorokine, J. Prilusky, E. E. Abola, and M. Edelman, “Automated analysis of interatomic contacts in proteins,” *Bioinformatics*, vol. 15, no. 4, pp. 327–332, 1999.
- [56] N. T. Doncheva, Y. Assenov, F. S. Domingues, and M. Albrecht, “Topological analysis and interactive visualization of biological networks and protein structures,” *Nat. Protoc.*, vol. 7, no. 4, pp. 670–685, Jan. 2012.
- [57] J. C. Biro and G. Fördös, “SeqX: a tool to detect, analyze and visualize residue co-locations in protein and nucleic acid structures,” *BMC Bioinformatics*, vol. 6, p. 170, Jan. 2005.
- [58] M. J. Pietal, I. Tuszynska, and J. M. Bujnicki, “PROTMAP2D: visualization, comparison and analysis of 2D maps of protein structure,” *Bioinformatics*, vol. 23, no. 11, pp. 1429–1430, Jan. 2007.
- [59] C. Vehlouw, H. Stehr, M. Winkelmann, J. M. Duarte, L. Petzold, J. Dinse, and M. Lappe, “CMView: Interactive contact map visualization and analysis,” *Bioinformatics*, vol. 27, no. 11, pp. 1573–1574, Jan. 2011.
- [60] M. Kayikci, A. J. Venkatakrishnan, J. Scott-Brown, C. N. J. Ravarani, T. Flock, and M. M. Babu, “Visualization and analysis of non-covalent contacts using the Protein Contacts Atlas,” *Nat. Struct. Mol. Biol.*, vol. 25, no. 2, pp. 185–194, Jan. 2018.
- [61] N. Atias and R. Sharan, “Comparative analysis of protein networks: hard problems, practical solutions,” *Commun. ACM*, vol. 55, no. 5, p. 88, Jan. 2012.
- [62] R. De Virgilio and S. E. Rombo, “Approximate matching over biological RDF graphs,” in *the 27th Annual ACM Symposium*, 2012, p. 1413.
- [63] N. Ferraro, L. Palopoli, S. Panni, and S. E. Rombo, “Asymmetric Comparison and Querying of Biological Networks,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 8, no. 4, pp. 876–889, Jan. 2011.
- [64] S. Panni and S. E. Rombo, “Searching for repetitions in biological networks:

- methods, resources and tools,” *Brief. Bioinform.*, vol. 16, no. 1, pp. 118–136, Jan. 2015.
- [65] R. Sharan, I. Ulitsky, and R. Shamir, “Network-based prediction of protein function,” *Mol. Syst. Biol.*, vol. 3, Jan. 2007.
 - [66] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, “Comparative assessment of large-scale data sets of protein–protein interactions,” *Nature*, vol. 417, no. 6887, pp. 399–403, Jan. 2002.
 - [67] E. Hartuv and R. Shamir, “A clustering algorithm based on graph connectivity,” *Inf. Process. Lett.*, vol. 76, no. 4–6, pp. 175–181, Jan. 2000.
 - [68] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, “Detection of functional modules from protein interaction networks,” *Proteins Struct. Funct. Bioinforma.*, vol. 54, no. 1, pp. 49–57, Jan. 2003.
 - [69] S. Tornow and H. W. Mewes, “Functional modules by relating protein interaction networks and gene expression,” *Nucleic Acids Res.*, vol. 31, no. 21, pp. 6283–6289, 2003.
 - [70] C. Pizzuti and S. E. Rombo, “Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods,” *Bioinformatics*, vol. 30, no. 10, pp. 1343–1352, Jan. 2014.
 - [71] D. Deb, S. Vishveshwara, and S. Vishveshwara, “Understanding Protein Structure from a Percolation Perspective,” *Biophys. J.*, vol. 97, no. 6, pp. 1787–1794, Jan. 2009.
 - [72] O. Gaci, “A Topological Description of Hubs in Amino Acid Interaction Networks,” *Adv. Bioinformatics*, vol. 2010, pp. 1–9, Jan. 2010.
 - [73] M. M. Teeter, “Water structure of a hydrophobic protein at atomic resolution: Pentagon rings of water molecules in crystals of crambin,” *Proc. Natl. Acad. Sci.*, vol. 81, no. 19, pp. 6014–6018, Jan. 1984.
 - [74] V. Batagelj and A. Mrvar, “Pajek — Analysis and Visualization of Large Networks,” in *Graph Drawing Software*, M. Jünger and P. Mutzel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 77–103.
 - [75] J. Bang-Jensen and G. Z. Gutin, *Digraphs*. London: Springer London, 2009.
 - [76] *Analysis of biological networks*. Hoboken, N.J: Wiley-Interscience, 2008.
 - [77] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertész, “Generalizations of the clustering coefficient to weighted complex networks,” *Phys. Rev. E*, vol. 75, no. 2, Jan. 2007.

- [78] H. Yu, P. M. Kim, E. Sprecher, V. Trifonov, and M. Gerstein, "The Importance of Bottlenecks in Protein Networks: Correlation with Gene Essentiality and Expression Dynamics," *PLoS Comput. Biol.*, vol. 3, no. 4, p. e59, Jan. 2007.
- [79] N. Biggs, E. K. Lloyd, and R. J. C. N.-Q. B. 1986 Wilson, *Graph theory, 1736-1936*. Oxford [Oxfordshire] ; New York: Clarendon Press, 1986.
- [80] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Jan. 1969.
- [81] J.-J. Tsay, B.-L. Wu, and Y.-S. Jeng, "Hierarchically organized layout for visualization of biochemical pathways," *Artif. Intell. Med.*, vol. 48, no. 2–3, pp. 107–117, Jan. 2010.
- [82] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical System Structures," *IEEE Trans. Syst. Man. Cybern.*, vol. 11, no. 2, pp. 109–125, Jan. 1981.
- [83] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exp.*, vol. 21, no. 11, pp. 1129–1164, Jan. 1991.
- [84] P. Eades, "A heuristic for graph drawing," vol. 42, pp. 149–160, 1984.
- [85] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, Jan. 1989.
- [86] E. M. Reingold and J. S. Tilford, "Tidier Drawings of Trees," *IEEE Trans. Softw. Eng.*, vol. SE-7, no. 2, pp. 223–228, Jan. 1981.
- [87] U. Dogrusoz, E. Giral, A. Cetintas, A. Civril, and E. Demir, "A layout algorithm for undirected compound graphs," *Inf. Sci. (Ny)*, vol. 179, no. 7, pp. 980–994, Jan. 2009.
- [88] G. Zhou and J. Xia, "OmicsNet: a web-based tool for creation and visual analysis of biological networks in 3D space," *Nucleic Acids Res.*, vol. 46, no. W1, pp. W514–W522, Jan. 2018.
- [89] M. Suderman and M. Hallett, "Tools for visually exploring biological networks," *Bioinformatics*, vol. 23, no. 20, pp. 2651–2659, Jan. 2007.
- [90] G. A. Pavlopoulos, A.-L. Wegener, and R. Schneider, "A survey of visualization tools for biological network analysis," *BioData Min.*, vol. 1, no. 1, Jan. 2008.
- [91] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A.-C. Gavin, "Visualization of omics data for systems biology," *Nat. Methods*, vol. 7, no. 3, pp. S56–S68, Jan. 2010.

- [92] G. Agapito, P. Guzzi, and M. Cannataro, “Visualization of protein interaction networks: problems and solutions,” *BMC Bioinformatics*, vol. 14, no. Suppl 1, p. S1, Jan. 2013.
- [93] G. W. Leong, S. C. Lee, C. C. Lau, P. Klappa, and M. S. S. Omar, “Comparison of Computational Tools for Protein-Protein Interaction (PPI) Mapping and Analysis,” *J. Teknol.*, vol. 63, no. 1, Jan. 2013.
- [94] M. Franz, C. T. Lopes, G. Huck, Y. Dong, O. Sumer, and G. D. Bader, “Cytoscape.js: a graph theory library for visualisation and analysis,” *Bioinformatics*, vol. 32, no. 2, pp. 309–311, 2015.
- [95] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, “Cytoscape 2.8: new features for data integration and network visualization,” *Bioinformatics*, vol. 27, no. 3, pp. 431–432, Jan. 2011.
- [96] M. Bastian, S. Heyyman, and M. Jacomy, “Gephi: an open source software for exploring and manipulating networks,” vol. 8, pp. 361–362, 2009.
- [97] S. D. Hooper and P. Bork, “Medusa: a simple tool for interaction graph analysis,” *Bioinformatics*, vol. 21, no. 24, pp. 4432–4433, Jan. 2005.
- [98] A. J. Enright and C. A. Ouzounis, “BioLayout—an automatic graph layout algorithm for similarity visualization,” *Bioinformatics*, vol. 17, no. 9, pp. 853–854, Jan. 2001.
- [99] K. R. Brown, D. Otasek, M. Ali, M. J. McGuffin, W. Xie, B. Devani, I. L. van Toch, and I. Jurisica, “NAViGaTOR: Network Analysis, Visualization and Graphing Toronto,” *Bioinformatics*, vol. 25, no. 24, pp. 3327–3329, Jan. 2009.
- [100] G. A. Pavlopoulos, S. I. O’Donoghue, V. P. Satagopam, T. G. Soldatos, E. Pafilis, and R. Schneider, “Arena3D: visualization of biological networks in 3D,” *BMC Syst. Biol.*, vol. 2, no. 1, p. 104, Jan. 2008.
- [101] “OpenAstexViewer.”
- [102] T. Nakane, “GLmol-Molecular Viewer on WebGL.” 2014.
- [103] S. Orchard, M. Ammari, B. Aranda, L. Breuza, L. Briganti, F. Broackes-Carter, N. H. Campbell, G. Chavali, C. Chen, N. del-Toro, M. Duesbury, M. Dumousseau, E. Galeota, U. Hinz, M. Iannuccelli, S. Jagannathan, R. Jimenez, J. Khadake, A. Lagreid, L. Licata, R. C. Lovering, B. Meldal, A. N. Melidoni, M. Milagros, D. Peluso, L. Perfetto, P. Porras, A. Raghunath, S. Ricard-Blum, B. Roechert, A. Stutz, M. Tognolli, K. van Roey, G. Cesareni, and H. Hermjakob, “The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases,” *Nucleic Acids Res.*, vol. 42, no. D1, pp. D358–D363, Jan. 2014.

- [104] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, M. Kuhn, P. Bork, L. J. Jensen, and C. von Mering, “STRING v10: protein–protein interaction networks, integrated over the tree of life,” *Nucleic Acids Res.*, vol. 43, no. D1, pp. D447–D452, Jan. 2015.
- [105] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, “DIP: the database of interacting proteins,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 289–291, 2000.
- [106] G. Joshi-Tope, “Reactome: a knowledgebase of biological pathways,” *Nucleic Acids Res.*, vol. 33, no. Database issue, pp. D428–D432, Jan. 2004.
- [107] K. R. Brown and I. Jurisica, “Online Predicted Human Interaction Database,” *Bioinformatics*, vol. 21, no. 9, pp. 2076–2082, Jan. 2005.
- [108] G. Csárdi and T. Nepusz, “The Igraph Software Package for Complex Network Research.” 2006.
- [109] S. Tisue and U. Wilensky, “NetLogo: A Simple Environment for Modeling Complexity,” p. 10.
- [110] M. A. Smith, B. Shneiderman, N. Milic-Frayling, E. Mendes Rodrigues, V. Barash, C. Dunne, T. Capone, A. Perer, and E. Gleave, “Analyzing (social media) networks with NodeXL,” in *the fourth international conference*, 2009, p. 255.
- [111] B. V Almende, “vis.js–A dynamic, browser based visualization library.” 2016.
- [112] “Ngraph.”.
- [113] “VivaGraphJS.”.
- [114] “Dagre.”.
- [115] M. Bostock, V. Ogievetsky, and J. Heer, “D³ Data-Driven Documents,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Jan. 2011.
- [116] “Springy.”.
- [117] “Alchemy.”.
- [118] A. Jacomy, “sigma.js.”.
- [119] “Dracula Graph Library.” 2010.
- [120] F. Kling, “JSNetworkX: A javascript port of the networkx graph library.” 2014.

- [121] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx,” Los Alamos National Lab. (LANL), Los Alamos, NM (United States), Jan. 2008.
- [122] J. H. Morris, C. C. Huang, P. C. Babbitt, and T. E. Ferrin, “structureViz: linking Cytoscape and UCSF Chimera,” *Bioinformatics*, vol. 23, no. 17, pp. 2345–2347, Jan. 2007.
- [123] M. Biasini, “PV-WebGL-based protein viewer.” 2014.
- [124] S. Bienert, A. Waterhouse, T. A. P. de Beer, G. Tauriello, G. Studer, L. Bordoli, and T. Schwede, “The SWISS-MODEL Repository—new features and functionality,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D313–D319, Jan. 2017.
- [125] *Graphical methods for data analysis*. Belmont, Calif. : Boston: Wadsworth International Group ; Duxbury Press, 1983.
- [126] S. Vijay-Kumar, C. E. Bugg, and W. J. Cook, “STRUCTURE OF UBIQUITIN REFINED AT 1.8 ANGSTROMS RESOLUTION,” (*:unav*), Jan. 1987.
- [127] L. Holm and C. Sander, “Mapping the Protein Universe,” *Science (80-.)*, vol. 273, no. 5275, pp. 595–602, Jan. 1996.
- [128] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph,” *Commun. ACM*, vol. 16, no. 9, pp. 575–577, Jan. 1973.
- [129] E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” *Theor. Comput. Sci.*, vol. 363, no. 1, pp. 28–42, Jan. 2006.
- [130] F. Cazals and C. Karande, “A note on the problem of reporting maximal cliques,” *Theor. Comput. Sci.*, vol. 407, no. 1–3, pp. 564–568, Jan. 2008.
- [131] M. Hulsmeier, M. T. Fiorillo, F. Bettosini, R. Sorrentino, W. Saenger, A. Ziegler, and B. Uchanska-Ziegler, “Crystal structure of HLA-B*2709 complexed with the vasoactive intestinal peptide type 1 receptor (VIPR) peptide (residues 400-408),” Jan. 2004.

APPENDIX

APPENDIX A. Pseudocodes for shortest path algorithms

```
1. function dijkstra(Graph, source)
2.   for each vertex v in Graph
3.     dist[v] := infinity
4.     previous[v] := undefined
5.   end for
6.   dist[source] := 0
7.   Q := the set of all nodes in Graph
8.   while Q is not empty
9.     u := vertex in Q with shortest distance[]
10.    if dist[u] = INFINITY
11.      break
12.    end if
13.    remove u from Q
14.    for each neighbour v of u
15.      alt := dist[u] + distance between u and v
16.      if alt < dist[v]
17.        dist[v] := alt
18.        previous[v] := u
19.        Remove key v from Q
20.      end if
21.    end for
22.  end while
23.  return dist[]
24. end
```

Figure A.1 Pseudocode for Dijkstra's shortest path algorithm.

```

1. for i=1 to N
2.   for j=1 to N
3.     if there is an edge from i to j
4.       dist[0][i][j] = the length of the edge from i to j
5.     else
6.       dist[0][i][j] = INFINITY
7.
8. for k=1 to N
9.   for i=1 to N
10.    for j=1 to N
11.      dist[k][i][j]=min(dist[k-1][i][j], dist[k-1][i][k] + dist[k-1][k][j])

```

Figure A.2 Pseudocode for Floyd-Warshall shortest path algorithm.

```

1. function bellmanFord(Graph, source)
2.   for each vertex V in Graph
3.     dist[V] <- INFINITY
4.     previous[V] <- NULL
5.   dist[source] <- 0
6.   for each vertex V in Graph
7.     for each edge (U,V) in Graph
8.       tempDistance <- dist[U] + edge weight between U and V
9.       if tempDistance < dist[V]
10.        dist[V] <- tempDistance
11.        previous[V] <- U
12.
13.   for each edge (U,V) in Graph
14.     if dist[U] + edge weight between U and V < dist[V]
15.       Error: Negative Cycle Exists
16.
17.   return dist[], previous[]

```

Figure A.3 Pseudocode for Bellman-Ford shortest path algorithm.

APPENDIX B. WebGL protein viewers

Table B.1. Comparison of the features of WebGL 3D protein structure viewers.

Library	Supported File Format	Representation (cartoon, lines, etc.)	Documentation	Custom Shape	Reference
GLMol	pdb, sdf, mol, mdl, xyz, ccp4	Yes	★☆☆	No	[129]
PV	pdf, sdf	Yes	★★★	Yes	[127]
NGL	pdb, sdf, mmCIF, cif, mcif, ent, pqr, gro, sdf, sd, mol2, mmtf,	Yes	★★★	Yes	[130]
3DMol.js	pdb, cdjson, json, cube, gro, mcif, cif, mmtf, mol2, pqr, prmtop, sdf, vasp, xyz	Yes	★★☆	Yes	[131]
Molmil	pdb, mol2, mdl, ccp4, gro, mmJSON, mmCIF	Yes	☆☆☆	No	[132]
Jolecule	pdb	Ribbon only	☆☆☆	No	[133]
Molvwr	pdb, mol, sdf, xyz	Spheres, balls and sticks, sticks	☆☆☆	No	[134]
three.js	pdb	Balls and sticks	★★☆	Yes	[135]
iview	pdb	Yes	★☆☆	No	[136]
Web3DMol	pdb	Yes	★★☆	No	[137]

AUTOBIOGRAPHY

Name Surname : Rasim Murat AYDINKAL
Place and Date of Birth : IZMIR /12.06.1988
Foreign Language : English
E-mail : a.rasimmurat@gmail.com

Education

Degree	Department/Program	University	GraduationYear
Bachelor of Science	Biology	Abant Izzet Baysal University (Bolu)	2011
Master of Science	Biology	Namik Kemal University (Tekirdag)	2016

Presentations:

1. Rasim Murat Aydinkal, Elife Zerrin Bagci, "Residue Packing at the Interfaces of Protein Complexes", 5th International BAU Drug Design Cong., Istanbul, 2017.
2. Rasim Murat Aydinkal, Onur Sercinoğlu, Pemra Ozbek Sarica, "Development of a web-based tool for network analysis of protein structures", International Conference on Applied Mathematics, Modeling and Life Sciences, Istanbul, 2018.

Article:

1. Rasim Murat Aydinkal, Elife Zerrin Bagci, (2018) “Residue packing in globular and intrinsically disordered proteins”, *Proteins: Structure, Function, and Bioinformatics*. doi: 10.1002/prot.25459

Thesis:

1. “Computational investigation of the extent of regularities in residue packing in a representative group of intrinsically disordered proteins”, *Namik Kemal University, Tekirdag*, 2016.