UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Grafi struktur proteinov: Uporaba teorije grafov za analizo makromolekulskih struktur

(Protein graphs: Using insights from Graph Theory in analysis of macromolecular structures)

Ime in priimek: Miloš Tomić
Študijski program: Matematika
Mentor: doc. dr. Jure Pražnikar

Koper, julij 2017

Ključna dokumentacijska informacija

Ime in PRIIMEK: Miloš TOMIĆ

Naslov zaključne naloge: Grafi struktur proteinov: Uporaba teorije grafov za analizo makromolekulskih struktur

Kraj: Koper

Leto: 2017

Število listov: 29 Število slik: 14 Število tabel: 1

Število referenc: 21

Mentor: doc. dr. Jure Pražnikar

Ključne besede: beljakovine, aminokisline, teorija grafov, grafi struktur proteinov, premer, drevo najkrajših poti, stopnja grafa, energija grafa

Math. Subj. Class. (2010): 92E10

Izvleček:
Beljakovine imajo ključno vlogo pri številnih biokemijskih procesih. Poznavanje strukture in funkcije beljakovin je torej ključnega pomena za proučevanje procesov v živih organizmih. Struktura proteina je zelo kompleksna in jo navadno predstavimo na tirih nivojih: 1) primarna struktura, kjer opišemo zaporedje aminokislin, 2) sekundarna struktura (alfa vijačnice, beta plošče in zanke), 3) terciarna struktura in 4) kvartarna struktura. Če želimo natančno analizirati funkcije proteinov je potrebno poznati njihovo 3D strukturo.
Da bi bolje analizirali in validirali 3D strukturo proteinov, smo 3D strukturo proteina predstavili v obliki grafa in analizirali značilnosti (premer, drevo najkrajših poti, stopnja grafa, energija grafa, ter druge) tako dobljenih grafov.
Pri analizi grafov smo posebno pozornost namenili 3D strukturam proteinov, ki so bile napačno rešene in kasneje popravljene.
Analiza grafov skonstruiranih iz 3D struktur proteinov je tako pokazala, da bi lahko značilnosti takšnih grafov uporabili kot alternativni, oziroma dodaten validacijski korak pri reševanju struktur proteinov.

Key words documentation

Name and SURNAME: Miloš TOMIĆ

Title of final project paper: Protein graphs: Using insights from Graph Theory in analysis of macromolecular structures

Place: Koper

Year: 2017

Number of pages: 29 Number of figures: 14 Number of tables: 1

Number of references: 21

Mentor: Assist. Prof. Jure Pražnikar, PhD

Keywords: proteins, amino acids, Graph Theory, protein graphs, radius of a graph, average shortest path, average node degree, graph energy

Math. Subj. Class. (2010): 92E10

Abstract:
It is well known that protein molecules play many critical roles in the nature. Finding a way to understand the chemical behavior of protein molecules, i.e. explaining their structure and interaction with other molecules, is a key to complete understanding how many biological processes work.
Protein structure is very complex and it is divided into 4 levels: primary, secondary, tertiary and quaternary. Although one protein is determined by its primary structure (sequence of amino acids that form it), secondary (position of local segments in 3D) and tertiary (protein folding) structures can vary. So, in theory, there are several 3D models of the same protein, but in nature, we can rarely find more than one macromolecular structure. Finding a correct structure of a protein is a key step in further analysis of protein behavior, but this process can be very demanding and expensive.
That kind of analysis will yield some important conclusions regarding the behavior of that specific protein. We will analyze graphs of those proteins with distinguished correct and incorrect 3D structure (that have been experimentally confirmed) and try to specify what values of graph properties should correct structures have, compared to those that are incorrect. Purpose of this paper is to show that analysis of protein graphs can be used as a helpful research tool in macromolecular modeling.

Acknowledgement

I would like to express deep gratitude to my mentor, Assist. Prof. Jure Pražnikar, PhD, for introducing me to this topic and for his selfless guidance during my work on this project paper.

I would like to thank the professors of Faculty of Mathematics, Natural Sciences and Information Technologies for the knowledge they handed me over, and to the technical staff of the Faculty for their help with all sorts of administrative procedures. Also, I am deeply grateful for the scholarship that University of Primorska provided me with during my studies.

I would like to thank my family, especially my mother, for the unconditional love and support in every aspect of my life.

In the end, I would like to thank Danijela, who was with me on every step of the way and without whom this journey would not be possible.

Contents

List of Tables

List of Figures

List of Abbreviations

i.e. that is
etc. and the rest
et al. and others
2D two dimensional
3D three dimensional

1 Introduction

It is well known that protein molecules play many critical roles in the nature. Finding a way to understand the chemical behavior of protein molecules, i.e. explaining their structure and interaction with other molecules, is a key to complete understanding how many biological processes work. One might say that if we know more about the proteins, we can also learn more about life on Earth itself.

Protein structure is very complex and it is divided into 4 levels: primary, secondary, tertiary and quaternary.

- *Primary protein structure* is based on the linear sequence of amino acids in a protein, starting from a amino-terminal and ending in a carboxyl-terminal. There are 20 essential or proteinogenic amino acids, listed below [16].
- *Secondary protein structure* is the three dimensional form of local segments of proteins. The two most common secondary structural elements are alpha helices and beta sheets [16].
- *Tertiary secondary structure* is the three dimensional shape of a protein. Secondary structure elements typically spontaneously form as an intermediate before the protein folds into its three dimensional tertiary structure [16].
- *Quaternary protein structure* refers to the number and arrangement of the protein subunits with respect to one another [16].

No.	Name	3-letter abbreviation
1.	Alanine	ALA
2.	Arginine	ARG
3.	Aspargine	ASN
4.	Aspartic acid	ASP
5.	Cysteine	CYS
6.	Glutamic acid	GLU
7.	Glutamine	GLN
8.	Glycine	GLY
9.	Histidine	HIS
10.	Isoleucine	ILE
11.	Leucine	LEU
12.	Lysine	LYS
13.	Methionine	MET
14.	Phenylalanine	PHE
15.	Proline	PRO
16.	Serine	SER
17.	Threonine	THR
18.	Tryptophan	TRP
19.	Tyrosine	TYR
20.	Valine	VAL

Table 1.1: List of essential amino acids

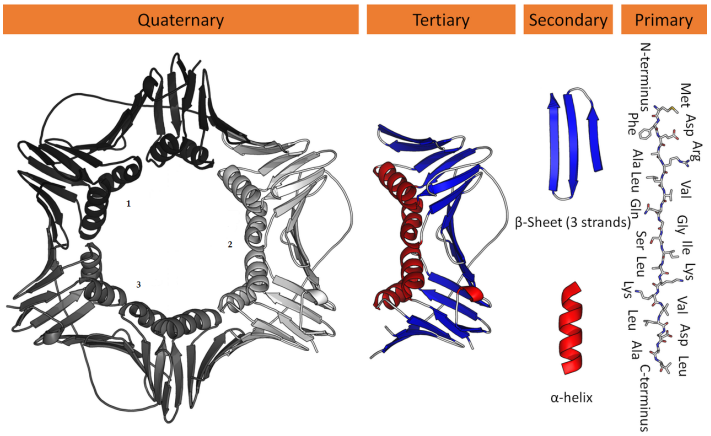

Figure 1.1: Four levels of protein structure using PCNA protein as an example [16]

Although one protein is determined by its primary structure (sequence of amino acids that form it), secondary (position of local segments in 3D) and tertiary (protein folding) structures can vary.
So, in theory, there are several 3D models of the same protein, but in nature, we can rarely find more than one macromolecular structure. Finding a correct structure of a protein is a key step in further analysis of protein behavior, but this process can be very demanding and expensive.
The determination of 3D macromolecular structures consists of experimental data recording and model building, refinement and validation. Often, the experimental data is not complete and can contain errors, and scientists can sometimes misinterpret the data, which can result in incorrect structure determination.

Graph Theory is a branch of discrete mathematics, distinguished by the geometric approach to the study of objects. The principal object of the theory is a graph and its generalization. Any problem or object under consideration is represented in the form of nodes (vertices, elements) and edges (connections) [13].
According to Vishveshwara et al. [13] : *'Although the topic is more than two centuries old, only in recent times it has gained momentum and has been routinely used in various branches of science and engineering. The mathematics developed earlier can now be applied to systems with large number of vertices and edges, since computers can be effectively made use of in obtaining solutions to such large graphs. Extensive applications of graph theory are made use of in the fields such as electrical circuits, communication and transportation networks.'*

In order to better analyze the 3D protein structure, we can transform the protein into a graph. Namely, a protein structure has geometry, expressed in the conformation of the protein backbone and side-chains.
Many structures can differ in terms of the conformational features (geometry) but still can have the same topology (the gross shape). Thus, it is best to transform the protein into a graph and examine its properties. That kind of analysis will yield some important conclusions regarding the behavior of that specific protein.

In this paper, the following properties of a protein graph will be considered:

- Average node degree
- Average shortest path
- Clustering coefficient

- Radius of a graph
- Eigenvector centrality (Largest eigenvalue of an adjacency matrix and the corresponding eigenvector)
- Energy of a graph
- Graph entropy and label entropy

We will analyze graphs of those proteins with distinguished correct and incorrect 3D structure (that have been experimentally confirmed) and try to specify what values of graph properties mentioned above should correct structures have, compared to those that are incorrect.

2 Construction of protein graphs

Definition 2.1. A graph $G = G(V, E)$ consists of a set of vertices (nodes) V and a set of edges E, in which the vertices and edges are related as follows: Two vertices v_i and v_j of a graph G are said to be adjacent if there is an edge e_{ij} connecting them.

Technical note: The vertices v_i and v_j are then said to be incident to the edge e_{ij}. Two distinct edges of a graph G are adjacent if they have at least one vertex in common.

Question that naturally arises is how to effectively transform a protein, that is a structure in 3D, into a 2D model of a graph. First, we have to introduce an effective way of representing proteins as lists. In this case, PDB (protein data bank) format will be used. Since 1971, the Protein Data Bank archive (PDB) has served as the single repository of information about the 3D structures of proteins, nucleic acids, and complex assemblies.
Protein Data Bank (PDB) format is a standard for files containing atomic coordinates. It is used for structures in the Protein Data Bank and is read and written by many programs. PDB format consists of lines of information in a text file. Each line of information in the file is called a record. A PDB file generally contains several different types of records, arranged in a specific order to describe a structure [17].
In the following picture, we can see an example of a PDB file. These are first 19 atoms (first 3 amino acids) of a protein glucagon.

```
ATOM      1  N   HIS A   1      49.668  24.248  10.436  1.00 25.00           N
ATOM      2  CA  HIS A   1      50.197  25.578  10.784  1.00 16.00           C
ATOM      3  C   HIS A   1      49.169  26.701  10.917  1.00 16.00           C
ATOM      4  O   HIS A   1      48.241  26.524  11.749  1.00 16.00           O
ATOM      5  CB  HIS A   1      51.312  26.048   9.843  1.00 16.00           C
ATOM      6  CG  HIS A   1      50.958  26.068   8.340  1.00 16.00           C
ATOM      7  ND1 HIS A   1      49.636  26.144   7.860  1.00 16.00           N
ATOM      8  CD2 HIS A   1      51.797  26.043   7.286  1.00 16.00           C
ATOM      9  CE1 HIS A   1      49.691  26.152   6.454  1.00 17.00           C
ATOM     10  NE2 HIS A   1      51.046  26.090   6.098  1.00 17.00           N
ATOM     11  N   SER A   2      49.788  27.850  10.784  1.00 16.00           N
ATOM     12  CA  SER A   2      49.138  29.147  10.620  1.00 15.00           C
ATOM     13  C   SER A   2      47.713  29.006  10.110  1.00 15.00           C
ATOM     14  O   SER A   2      46.740  29.251  10.864  1.00 15.00           O
ATOM     15  CB  SER A   2      49.875  29.930   9.569  1.00 16.00           C
ATOM     16  OG  SER A   2      49.145  31.057   9.176  1.00 19.00           O
ATOM     17  N   GLN A   3      47.620  28.367   8.973  1.00 15.00           N
ATOM     18  CA  GLN A   3      46.287  28.193   8.308  1.00 14.00           C
ATOM     19  C   GLN A   3      45.406  27.172   8.963  1.00 14.00           C
```

Figure 2.1: PDB representation of glucagon [17]

Notice that each line or record begins with the record type ATOM. Although there

are other types, such as SHEET, HELIX, TER etc, in this paper, only ATOM record type will be considered.
The atom serial number is the next item in each record.
The atom name is the third item in the record. Notice that the first one or two characters of the atom name consists of the chemical symbol for the atom type. All the atom names beginning with C are carbon atoms; N indicates a nitrogen and O indicates oxygen. In amino acid residues, the next character is the remoteness indicator code, which is transliterated according to: $\alpha = A, \quad \beta = B, \quad \gamma = G, \quad \delta = D, \quad \epsilon = E, \quad \zeta = Z, \quad \eta = H$.
The next character of the atom name is a branch indicator, if required.
The next data field is the residue type. Notice that each record contains the residue type. In this example, the first residue in the chain is HIS (histidine) and the second residue is a SER (serine).
The next data field contains the chain identifier, in this case A.
The next data field contains the residue sequence number. Notice that as the residue changes from histidine to serine, the residue number changes from 1 to 2. Two like residues may be adjacent to one another, so the residue number is important for distinguishing between them.
The next three data fields contain the X, Y, and Z coordinate values, respectively.
The last three fields shown are the occupancy, temperature factor (B-factor), and element symbol(these quantities won't be used in the rest of the text).
The glucagon data file continues in this manner until the final residue is reached:

```
ATOM    239  N   THR A  29       3.391  19.940  12.762  1.00 21.00           N
ATOM    240  CA  THR A  29       2.014  19.761  13.283  1.00 21.00           C
ATOM    241  C   THR A  29       0.826  19.943  12.332  1.00 23.00           C
ATOM    242  O   THR A  29       0.932  19.600  11.133  1.00 30.00           O
ATOM    243  CB  THR A  29       1.845  20.667  14.505  1.00 21.00           C
ATOM    244  OG1 THR A  29       1.214  21.893  14.153  1.00 21.00           O
ATOM    245  CG2 THR A  29       3.180  20.968  15.185  1.00 21.00           C
ATOM    246  OXT THR A  29      -0.317  20.109  12.824  1.00 25.00           O
TER     247      THR A  29
```

Figure 2.2: End of glucagon PDB file [17]

Now, when we have an appropriate way of representing proteins, we proceed to construction of protein graphs.
We see that every residue in a protein has its X, Y, Z coordinates, so we can calculate a distance between any two residues.

Definition 2.2. *Euclidian distance in 3D*
For any two points P and Q represented in $3D$ space with X, Y, Z coordinates $P(x_1, y_1, z_1)$ and $Q(x_2, y_2, z_2)$ we can calculate Euclidian distance $D(P, Q)$ as:

$$D(P,Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

The main idea behind construction of protein graphs is to take amino acid residues as vertices (nodes). We abstract every amino acid to its $C\alpha$ (alpha carbon) atoms and measure the Euclidian distance for each pair of them. If the distance fits some threshold value, then we create an edge between the corresponding nodes.
Based on the research of Greene and Higman [5] threshold value can be set to several different values, but the best results were obtained when they used threshold value of $7\mathring{A}$ ($7\mathring{A}$(angstroms)=700 picometers). Thus, we obtain the following algorithm for construction of protein graphs:

Algorithm for construction of protein graphs
Input: Protein in PDB format
Idea: Obtain a protein graph of a given protein by taking $C\alpha$ atoms of every amino acid residue to serve as graph vertices(nodes) and assigning edges between two nodes if the distance of corresponding atoms is less or equal than threshold value.
Initialization: $V(G) = \emptyset$, $E(G) = \emptyset$
Iteration: If atom at position i is $C\alpha$ (CA) then $C\alpha_i \in V(G)$. For every $j < i$ s.t. $C\alpha_j \in V(G)$, we check if $D(C\alpha_i, C\alpha_j) \leq 7\mathring{A}$, then $e_{ij} \in E(G)$, that is, we construct an edge between them, i.e. we consider those two nodes to be adjacent.

3 Characterization and properties of protein graphs

Definition 3.1. A graph G is a *simple graph* if it has no loops (edges that start and end at the same vertex) and no multiple edges between vertices.

From the definition above and the algorithm given in the previous chapter, we clearly see that protein graphs are simple graphs. This is an important characteristic of protein graphs that will yield some other interesting properties later.

3.1 Average node degree and clustering coefficient

Definition 3.2. *The degree of a node* v, denoted $d(v)$, represents the number of nodes adjacent to v.

Definition 3.3. *The average node degree of a graph* G is the mean value of all node degrees in G. Formally written:

$$AND(G) = d(G) = \frac{1}{N}\sum_{i=1}^{N} d(v_i),$$

where $d(v_i)$ represents the degree of the node v_i and N is the total number of nodes in the graph G.

Remark 3.4. Another way of expressing the average node degree is with the ratio:

$$AND(G) = \frac{2e(G)}{N(G)},$$

where $e(G)$ represents the total number of edges in a graph G and $N(G)$ is a number of nodes in a graph G.
It also holds:

$$\delta(G) \leq AND(G) \leq \Delta(G),$$

where $\delta(G)$ and $\Delta(G)$ represent minimum and maximum node degree in graph G, respectively.

Definition 3.5. *The clustering coefficient of a node* v, denoted by $c(v)$, represents the completeness of the neighborhood of the node v. That is,

$$c(v) = \frac{2e_v}{k(v)(k(v)-1)},$$

where $k(v)$ represents the number of neighbors of node v and e_v is the number of neighboring nodes that are adjacent to each other.

Remark 3.6. If graph is simple, then $k(v) = d(v)$, for every node $v \in V(G)$.

Remark 3.7. If all the neighbor nodes of v are connected, then the neighborhood of v is complete and we have a clustering coefficient equal to 1. If no nodes in the neighborhood of v are connected, then the clustering coefficient is 0.

Definition 3.8. Clustering coefficient of the whole graph G on N vertices can be expressed as the mean value of clustering coefficients of all vertices:

$$C(G) = \frac{1}{N}\sum_{i=1}^{N} c(v_i)$$

Example 3.9.

Definition 3.10. *A complete graph* K_n is a simple graph in which every pair of distinct vertices is connected by a unique edge.

All nodes of K_n are of the same degree: $n-1$. Thus, $\delta(K_n) = AND(K_n) = \Delta(K_n) = n-1$.

It follows from definition of K_n that the neighborhood of every vertex is complete, so every vertex has clustering coefficient equal to 1. Therefore, $C(K_n) = 1$.

Example 3.11. Let us look at an another simple example:

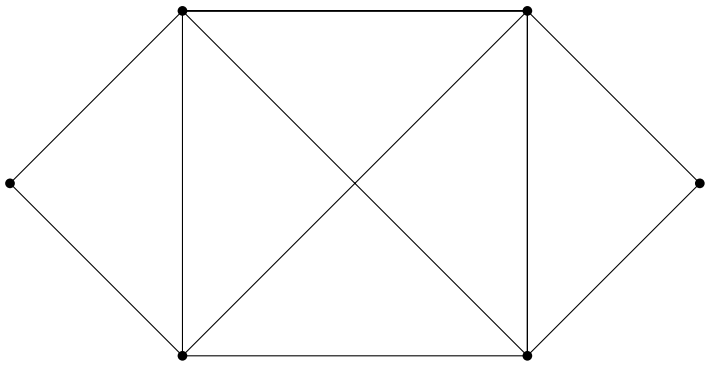

Figure 3.1: Example of a graph on 6 vertices

This graph has 6 vertices: two of them are of degree 2 and the other four are of degree 4.
So, the average node degree of this graph is:

$$AND(G) = \frac{2 \cdot 2 + 4 \cdot 4}{6} = 3,333$$

Vertices of degree 2 have a complete neighborhood, so they will have clustering coefficient equal to 1. Vertices of degree 4 don't have a complete neighborhood, so we have to use the formula. Let v be the one of those vertices. Since we are dealing with a simple graph $k(v) = d(v) = 4$ and we see from the picture that $e_v = 4$ (we count how many edges are there between neighbors of v).
So, $c(v) = \frac{2 \cdot 4}{4 \cdot 3} = \frac{2}{3}$.
In the end,

$$C(G) = \frac{2 \cdot 1 + 4 \cdot \frac{2}{3}}{6} = \frac{14}{18} = 0.778$$

When we look at the average node degree and the clustering coefficient of protein graphs, one of the first questions that arises is:
Do average node degree and clustering coefficient of protein graphs strongly depend on protein length and folding mechanism?
As we can see from the formula above, the average node degree is proportional to $1/N$, where N represents the number of nodes i.e. the length of the protein.
Protein folding will change the number of edges in a graph of a protein that we are looking into, because some amino acids will get close enough, so that we can say they are connected. Since the average node degree is proportional to $2e(G)$, we can say that it depends on the folding mechanism of a protein.
So, we can say that average node degree depends on both protein length and folding mechanism, but we can't say it 'strongly' depends on either one of them, because in both cases, the dependence is linear.
The same goes for the clustering coefficient. It is proportional to $1/N$ and e_v depends on $e(G)$. In the case of simple graphs (which protein graphs are), higher e_v yields higher $e(G)$ and vice versa. Therefore, we can say that clustering coefficient depends on protein length and folding mechanism, but that dependence is not 'strong'.

This was sort of a theoretical approach to this problem, but in our research we wanted to show that this actually holds in real examples of protein graphs.
We have obtained and analyzed the total of 50249 proteins from Protein Graph repository [4] [19]. All computations were conducted using NAPS: Network Analysis of

Protein Structures software, available online [3] [18].

It can be seen from the plots below that the slopes for both average node degree and clustering coefficient are rather flat, and that:

$$AND \propto N^{0.0234}$$

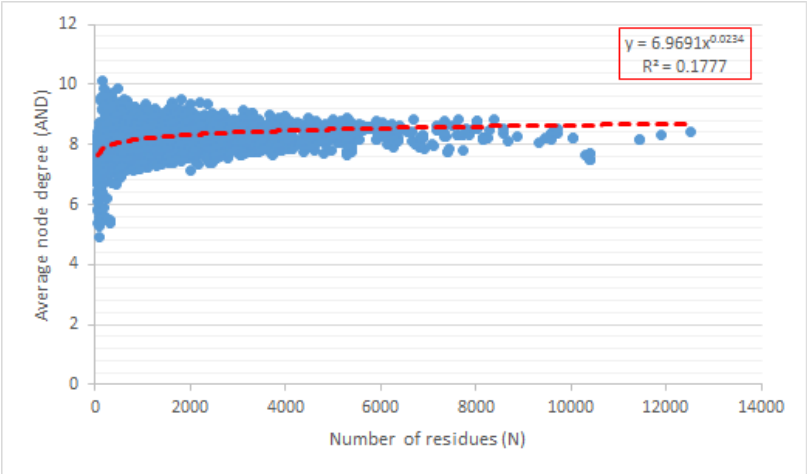

Figure 3.2: Dependence of protein length(number of residues) and average node degree of protein graphs

$$C \propto N^{-0.027}$$

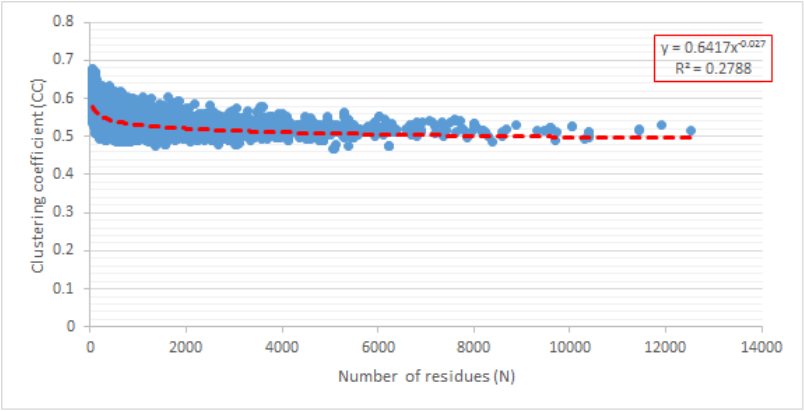

Figure 3.3: Dependence of protein length(number of residues) and clustering coefficient of protein graphs

The resulting scaling exponents of 0.023 and -0.028 for a set of structures in our study suggest that average node degree and clustering coefficient of protein graphs

are not strongly related with the protein size. Increasing the protein size 1000 times increases the average node degree and decreases the clustering coefficient by approximately 20% on average.

Thus, we can conclude that the average node degree and the clustering coefficient of proteins with diverse primary structure and of similar sizes are not randomly scattered over a wide range, but rather are distributed in a narrow interval.

Based on the given data, we can express average node degree in terms of protein size with:

$$AND \approx 7N^{0.023}$$

If we have had used $'log - log'$ plot, the expression would transform to $\ln(AND) \approx 0.023 \ln(N) + 1.94$, where we approximated $e^{1.94} \approx 7$.

We can also express clustering coefficient in terms of protein size with:

$$C \approx 0.64N^{-0.27},$$

or transform it to $\ln(C) = -0.027 \ln(N) - 0.44$.

Note that the standard deviation of these computations is $\sigma(AND) \approx 0.3$ and $\sigma(C) \approx 0.02$.

It is also worth noting that larger proteins have higher average node degree and lower clustering coefficient. Large proteins are made of several domains (subunits), thereby they form quaternary structure and hence establish additional edges in protein graphs that correspond to adjacent domains in $3D$ space. These new edges increase degree of some nodes and because of that AND increases, but they don't necessarily impact the completeness of neighborhoods of those nodes, so clustering coefficient decreases.

3.2 Average shortest path

Definition 3.12. Let $G = (V, E)$ be a graph possessing n vertices and m edges, with the set of vertices $V = \{v_1, v_2, \dots v_n\}$ and the edges set $E = \{e_1, e_2, \dots e_m\}$.
The *adjacency matrix* $A(G) = (a_{i,j})$ of G is the $n \times n$ matrix defined by:

$$a_{i,j} = \begin{cases} 1, & \texttt{if}(v_i, v_j) \in E \\ 0, & \texttt{otherwise} \end{cases}$$

Example 3.13. Adjacency matrix of a graph from Example 3.11 looks like:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Remark 3.14. Since protein graphs are simple graphs, then we see that adjacency matrix of any protein graph is *symmetric, with zeros on the diagonal.*

Definition 3.15. The *shortest path* between two nodes v_i and v_j in a graph G is the minimal number of edges that lie between two given nodes.

In computing the shortest path between a pair of nodes, we make use of the fact that the number of different paths connecting a pair of nodes in n steps is given as the entry of a adjacency matrix to the power of n, i.e. $B_{ij} = (A^n)_{ij}$. So, the shortest path L_{ij} between nodes v_i and v_j is the minimal power m of A for which $(A^m)_{ij}$ is nonzero [1].

Definition 3.16. The *average shortest path* of the graph G on N vertices is then given by the formula:

$$L = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} L_{ij}$$

Another way to obtain shortest path between vertices is to use *Dijkstra's Algorithm* [20]. This algorithm is usually applied to weighted graphs, but it can be also useful when dealing with protein graphs (which are unweighted), by assigning weight 1 to every edge of the graph and weight inf if there is no edge between two vertices.

Let u be the initial vertex. The main idea is to maintain a set S of vertices to which a shortest path from u is known, enlarging S to include all vertices. To do this, we maintain a tentative distance $t(z)$ from u to each vertex z that is not in S, where $t(z)$ is the length of a shortest u, z-path found so far.

In every step, we select a vertex v outside S such that $t(v)$ is minimal, and add v to S. Then, we explore edges from v to update tentative distances: for each edge vz with z not belonging to S, we update $t(z) = \min\{t(z), t(v) + 1\}$. The iteration continues until $S = V(G)$, i.e. until we include all vertices or until $t(z) = \inf$ for every z not already included in S.

We say that a network, such as protein graph, exhibits *'small-world' properties* if it has high clustering coefficient and if its average shortest path scales logarithmically with the number of nodes, as shown in [5] [2] [1].
This basically means that two nodes don't necessarily have to be connected, but they most likely share a neighbor, that is, they can be reached in fairly small number of steps. This further implies that protein structures should have average shortest path as small as possible.

3.3 Radius of a graph

Definition 3.17. *Radius* of the graph G, written $rad(G)$, is the minimal eccentricity, where *eccentricity* represents the maximal shortest path in the graph G.

In our analysis, we concluded that radius of protein graphs shows following dependence regarding the number of residues (N)

$$rad(G) \propto N^{\nu}, \quad \nu \approx 2/5$$

Hong and Lei in [7] tried to compare radius of a graph and radius of gyration.
Radius of gyration refers to distribution of the components of an object around an axis. Intuitively, we see that radius and radius of gyration have to be related to each other,that is, we can observe the axis of gyration, as well as all components of the protein, as nodes, so the radius of gyration in that case will correspond to radius of a graph (maximal shortest path between 'axis' node and other nodes).
They have mathematically calculated that: $R_g \propto N^{\nu}$ where N represents the number of residues and ν is an exponent depending on the solvent conditions.

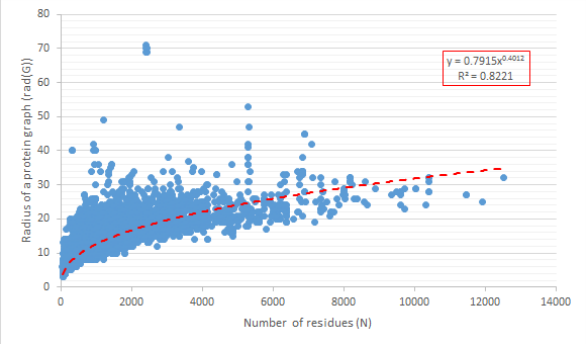

Figure 3.4: Dependence of protein length(number of residues) and radius of protein graphs

After some additional calculations, based on Flory theory, they've obtained the following correlation:

$$\nu = \frac{\alpha + 2}{5\alpha},$$

where α represents the fractional dimension of protein conformation. In a good solvent $\alpha \approx 1$ and $\nu = \frac{3}{5}$, while in a poor solvent conditions $\alpha \approx 3$ and $\nu = \frac{1}{3}$. Natural proteins under physiological conditions have $\alpha \approx 2$ and $\nu = \frac{2}{5}$.
Furthermore, authors of the paper used least squares method on 37162 proteins in PDB they've analyzed, and obtained numerically $\nu \approx 0.3915$ as the best fitting. This coincides with our result from Figure 3.4 ($\nu \approx 0.4012$).
Numerically obtained ν and ν obtained theoretically will be the same when we deal with natural proteins under physiological conditions i.e. proteins that occur in living organisms, which mostly is the case.
Also, we can note that it follows from the definition of a graph radius, that it is the minimal value taken from all maximal shortest paths in a graph (for every node, we look for a maximal shortest path to some other node, and then take minimum of those values). Thus, we conclude that radius of a graph should be as small as possible and that the correct structure of a certain protein will have smaller radius, compared to the incorrect structure of the same protein.

3.4 Largest eigenvalue (LEV) and corresponding eigenvector (eigenvector centrality)

Let $G = (V, E)$ be a graph possessing n vertices and m edges, with the set of vertices $V = v_1, v_2, , v_n$ and the edges set $E = e_1, e_2, , e_m$

As stated before, adjacency matrix of a simple graph is symmetric, and therefore has a complete set of real eigenvalues and an orthogonal eigenvector basis. The set of eigenvalues of a graph is the spectrum of the graph. We will denote the eigenvalues as $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_N$.

We obtain eigenvalues as the roots of the characteristic polynomial of matrix A, that is, we look for the solutions of the equation:

$$\det(A - \lambda I) = 0,$$

where I is the identity matrix. Eigenvalues of adjacency matrix also fulfill the following two conditions:

$$\sum_{i=1}^{n} \lambda_i = 0,$$

and

$$\sum_{i=1}^{n} \lambda_i^2 = 2m,$$

where m is total number of edges in the graph.
For every eigenvalue, we can find at least one vector $\vec{x}$ for which it holds:

$$\lambda\vec{x} = A\vec{x}$$

Vector $\vec{x}$ is called a corresponding eigenvector of the given eigenvalue.

The Perron-Frobenius theorem [21], asserts that a real square matrix with positive entries has a unique largest real eigenvalue and that the corresponding eigenvector can be chosen to have strictly positive components. Let $\lambda_1 = lev$.

Remark 3.18. The largest eigenvalue (lev) depends upon the highest degree in the graph. For any *k-regular* graph G (a graph with k degree on all vertices), the eigenvalue with the largest absolute value is k. Similarly, we can say that the lev of *clique* on N vertices (clique is a graph with all pairwise adjacent vertices) is $N - 1$.
In an irregular graph, lev is bounded with minimal and maximal node degree: $\delta(G) \leq lev \leq \Delta(G)$.

Another property we can use is *eigenvector centrality*. In graph theory, eigenvector centrality (also called eigencentrality) is a measure of the influence of a node in a network.

The relative centrality score of vertex v can be defined as:

$$x_v = \frac{1}{\lambda} \sum_{t \in N(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t,$$

where $N(v)$ is the set of neighbors of v, $a_{v,t}$ is an entry from the adjacency matrix (it is 1 if vertices v and t share an edge and 0 otherwise), and λ is a constant, that is actually going to be the eigenvalue. Namely, with a small rearrangement this expression, can be rewritten in vector notation as the eigenvector equation defined above:

$$\lambda \vec{x} = A\vec{x}$$

In general, there will be many different eigenvalues λ for which a non-zero eigenvector solution exists. However, the additional requirement that all the entries in the eigenvector be non-negative implies (by the Perron-Frobenius theorem) that only the greatest eigenvalue results in the desired centrality measure. The v^{th} component of the related eigenvector then gives the relative centrality score of the vertex v in the network.

In protein graphs, eigenvector centrality can be particularly useful when we want to examine which amino acids in the protein have the biggest impact on the stability of the protein, which side of a protein is the acive one in various chemical processes etc.

Example 3.19. Protein Cathepsin H(8PCB) has an eight residue long propeptide termed mini-chain with a disulfide bond link to the main-chain. There are two alternative directions to the mini-chain and only one of them is correct. The positions only differ for approximately 180 degrees (the mini-chain can be oriented 'up' or 'down'). Mini-chain takes positions from 221-228 and if we compare eigenvector centrality of the correct structure with the incorrect one, we see from the plot that the correct structure has a 'spike' at the end, that is, values of eigenvector centrality for a correct position of a mini-chain are substantially larger than those of the incorrect one.

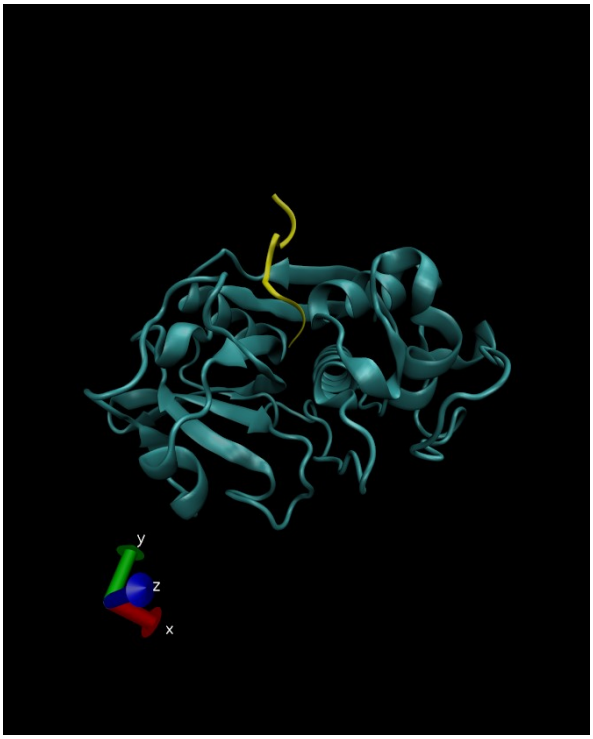

Figure 3.5: 3D representation of Cathepsin H (8PCB) with mini-chain (yellow) in the correct position

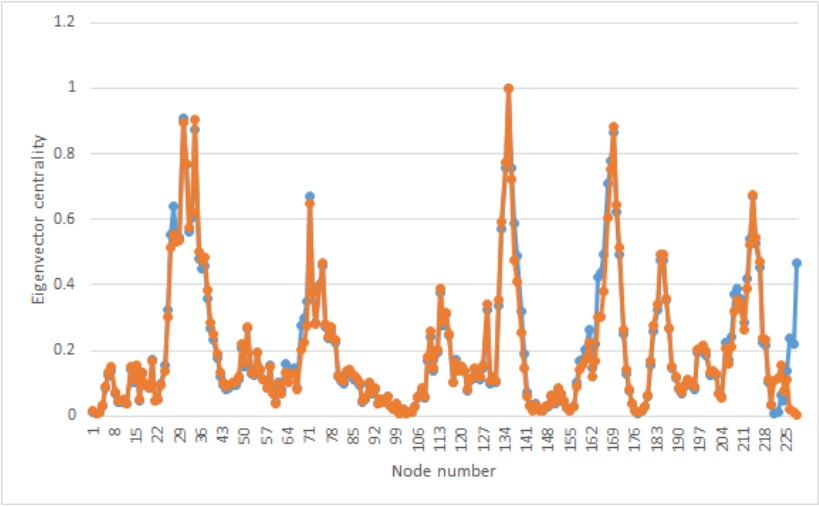

Figure 3.6: Eigenvector centrality for Cathepsin H (8PCB) with correct (blue) and incorrect (orange) position of the mini-chain

3.5 Energy of a graph

Definition 3.20. Let G be a graph and let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be the eigenvalues of adjacency matrix $A(G)$.
The *graph energy* $E(G)$ of G is defined as:

$$E(G) = \sum_{i=1}^{N} |\lambda_i|$$

Graph energy is an important criterion in analysis of protein graphs, because it shows the stability of connections in the graph/network.
Another approach that Wu et al. [15] suggest, is that every amino acid is a graph in its own right, i.e. it can be represented by a graph, where codons (triples of nucleotide bases A, C, G and T) serve as nodes, and thus it has its own graph energy. Graph energy of the 20 proteinogenic amino acids is determined and given in the list below.

Amino acid	E(G)
Ala	15.8276
Cys	7.6506
Asp	7.9136
Glu	8.246
Phe	10.198
Gly	20.2298
His	8.2288
Ile	16.1246
Lys	10.198
Leu	24.688
Met	4.4722
Asn	10.198
Pro	20.2298
Gln	8.246
Arg	28.011
Ser	20.1436
Thr	14.6968
Val	16.1488
Trp	4.4722
Tyr	10.198

Figure 3.7: Graph energy of proteinogenic amino acids

Given a protein sequence $S = S_1, S_2, , S_N$ the graph energy of the protein is defined as follows:

$$E(S) = \sum_{i=1}^{N} E(S_i),$$

where S_i represents the $i - th$ amino acid in the protein sequence S.
We see that the values of graph energy in this method differ from the first one, but

this alternative approach can have its advantages and it is primarily used with some other mathematical methods for looking for similarities/dissimilarities of two possibly the same protein structures [15].

If go back to the first method of computing graph energy, we see that it depends on the protein length, because adjacency matrix will have N eigenvalues, so $E(G) \propto N$. Therefore, we can use this graph property to do a more detailed analysis of protein structures we are interested in.

3.6 Label entropy and graph entropy

Entropy defines a quantitative equilibrium property within a system and it implies the principle of disorder, i.e. it is a measure of disorder of the system [9].
Although graph entropy and label entropy may seem to be similar, there is a significant difference: Label entropy represents diversity in the node labels (in case of protein graphs, those are types of amino acids) and the graph entropy is considering node degree.

Definition 3.21. *Label entropy*, also known as Shannon information entropy, measures the uncertainty of labels. Label entropy is given by the formula:

$$H = -\sum_{i=1}^{M} P_i \log_2(P_i),$$

where P_i is the fraction of residues of amino acid type i and M is the number of amino acid types (20 proteinogenic amino acids).
H ranges from 0 (only one residue in present at that position) to 4.322 (all 20 residues are equally represented in that position).

Technical note: Some sources use natural logarithm instead of $\log_2$, which would yield maximal entropy equal to 2.9957.

In our research, when we plotted label entropy, it was clear that it does not scale with number of residues (N) ($H \propto N^{0.0093}$), so it may be not the best tool for distinguishing between correct and incorrect protein structures.

On the other hand, for an n-object system, such as graph G, we can define the *graph entropy* in the following way:

$$I(G) = \sum_{k} -P(k) \log_2(P(k)),$$

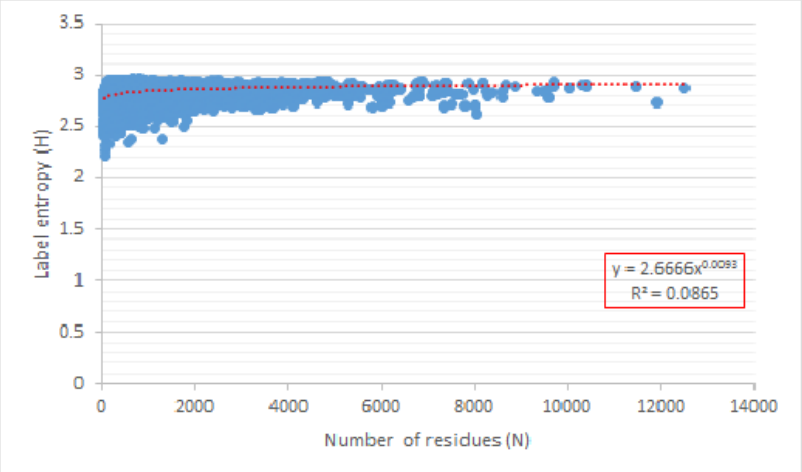

Figure 3.8: Number of residues(N)-Label entropy(H) plot

where $P(k)$ represents the probability that a node of the graph will have a degree k. It is usually hard to work with these probabilities and some adjustments are necessary for the graph entropy,proteins are so called scale-free networks [5] [9] i.e. the probabilities in this case follow the so-called *power law*:

$$P(k) = Ak^{-\gamma},$$

where γ is an exponent which usually lie in the interval $2 < \gamma < 3$ and A is a constant that insures $P(k)$ is less than 1.
If we plug in this formula into the previous one, we obtain:

$$I(G) = \sum_{k} -Ak^{-\gamma} \log_2(Ak^{-\gamma})$$

We want to find minimum and maximum entropy for the proteins. Since $I(G)$ depends on γ and $2 < \gamma < 3$, we will obtain maximum by taking $\gamma = 2$ and let $k \to \infty$. Thus, we will get a converging sum

$$-\sum_{k\to\infty} \frac{\ln(\frac{1}{k^2})}{k^2 \ln(2)} \approx 2,705$$

We repeat the process with $\gamma = 3$ and obtain:

$$-\sum_{k\to\infty} \frac{\ln(\frac{1}{k^3})}{k^3 \ln(2)} \approx 0,857$$

Note that A is omitted, because it is just a scaling constant. So, $0,857 < I(G) < 2,705$
Entropy of a graph highly depends on the number of edges, but not so much on the

number of nodes (protein length).

Therefore, both label entropy and graph entropy can't be used as a valid proof of correctness/incorrectness of a certain protein structure.

3.7 Some additional examples

In the figure below, there are 16 structures, paired up, such that in each pair one structure is correct and the other one is incorrect.
It was experimentally confirmed that correct structures are: *2phy, 2frh, 3pte, 3b5d, 3enl, 5fd1, 1xya.*

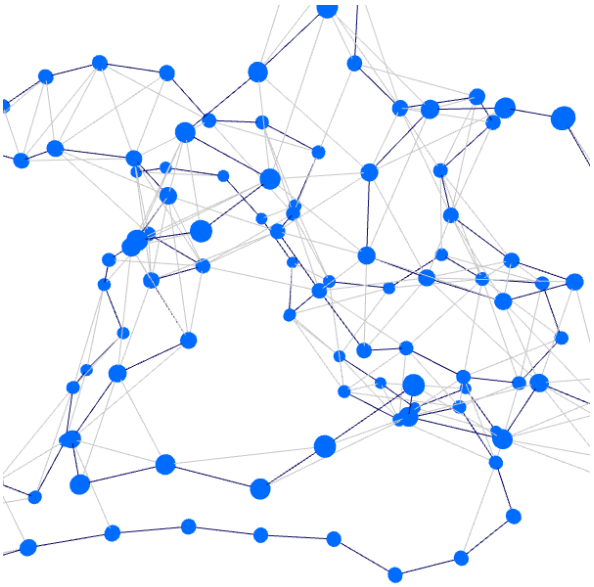

Figure 3.9: 2fd1 protein graph

Figure 3.10: 2fd1 protein graph

If we take one pair, say, *2fd1* and *5fd1*, and look at their respective graphs, we can already see that the correct structure *5fd1* seems better intra-connected, with better distribution of edges. So, by just visually examining the corresponding protein graphs, we can intuitively conclude, which structure is the correct one. But, what do values of graph properties tell us?
In the figure below, we clearly see that the correct structures, when compared to the incorrect ones, have:

- Larger average node degree
- Larger graph energy
- Smaller average shortest path
- Smaller or equal radius of a graph

Pairs	Nodes	avg. ND	Energy	ShortP	Radius	avg. ND	Energy	ShortP	Radius
1phy-2phy	125	7.02	254.8	4.41	7	7.73	261.6	4.1	5
1fzn-2frh	116	7.12	228.9	5.62	9	7.57	234	4.65	6
1pte-3pte	334	7.48	694.7	5.98	7	8.27	720.9	5.62	7
2f2m-3b5d	198	7.21	384.9	8.62	12	7.96	410.4	5.28	7
1enl-3enl	436	8.05	931.9	6.37	8	8.52	955.3	6.27	8
2fd1-5fd1	106	5.17	192.1	4.7	6	7.57	222.5	3.54	4
2hvp-3hvp	92	6.59	182.1	4.32	5	7.48	187.9	3.9	6
3xia-1xya	377	7.17	766.8	7.61	12	7.76	785.3	7.5	11

Figure 3.11: 8 pairs of correct and incorrect protein structures with their protein graph properties

Thus, we can say that these properties of protein graphs fully validate experimental data.

4 Conclusion

Today, the determination of macromolecular models still requires human interpretation of experimental data, and we should be aware of the occurrence of incomplete data during model building and refinement.
On the other hand, analysis of protein graphs does not relay on experimental data (it is purely theoretical), so it offers an orthogonal approach to interpretation and validation of protein structures. During our research, we have shown which properties of graph proteins will yield a clear difference between correct and incorrect structures in most of the cases. We will be able to recognize correct protein structures, because they will exhibit:

- Larger average node degree
- Larger graph energy
- Smaller average shortest path
- Smaller or equal radius of a graph,

when compared to the incorrect structures.
Therefore, we can consider property analysis of graph proteins to be an alternative approach and a helpful research tool in macromolecular modeling.

5 Povzetek naloge v slovenskem jeziku

Beljakovine imajo ključno vlogo pri številnih biokemijskih procesih. Poznavanje strukture in funkcije beljakovin je torej ključnega pomena za proučevanje procesov v živih organizmih. Struktura proteina je zelo kompleksna in jo navadno predstavimo na tirih nivojih: 1) primarna struktura, kjer opišemo zaporedje aminokislin, 2) sekundarna struktura (alfa vijačnice, beta plošče in zanke), 3) terciarna struktura in 4) kvartarna struktura. Če želimo natančno analizirati funkcije proteinov je potrebno poznati njihovo 3D strukturo. Najbolj pogosta metoda za določevanje strukture proteinov je rentgenska praškovna difrakcija. Osnovni koraki pri praškovni difrakciji so kristalizacija, izvajanje meritev, oziroma pridobivanje eksperimentalnih podatkov, gradnja in izboljšava modela, ter validacija. Eksperimentalni podatki, ki jih pridobimo tekom snemanja difrakcijskih slik, vsebujejo napake in niso popolni. Zato je pomembno, da model validiramo in tako lahko z večjo verjetnostjo potrdimo pravilnost modela.

Da bi bolje analizirali in validirali 3D strukturo proteinov, smo 3D strukturo proteina predstavili v obliki grafa in analizirali značilnosti (premer, drevo najkrajših poti, stopnja grafa, energija grafa, ter druge) tako dobljenih grafov. Graf iz 3D modela proteina je bil skonstruiran na naslednji način: vsak $C\alpha$ atom predstavlja vozlišče in e je razdalja med katerima koli $C\alpha$ atomoma manjša ali enaka $7\mathring{A}$, potem naredimo povezavo med dvema vozliščema.

Pri analizi grafov smo posebno pozornost namenili 3D strukturam proteinov, ki so bile napačno rešene in kasneje popravljene. Na teh modelih (napačno/pravilno) smo ugotovili, da imajo grafi pravilno rešenih struktur višjo energijo, krajšo najkrajšo pot in višjo stopnjo povezanosti. Intuitivno lahko rečemo, da so pravilno rešeni modeli bolj robustni, ter da se po grafu pravilno rešenega modela informacije pretakajo bolj učinkovito. Prav tako je analiza več kot 50000 grafov skonstruiranih iz 3D struktur proteinov pokazala, da stopnja povezanosti ni odvisna od primarne strukture in je zelo malo odvisna od velikosti-dolžine proteina.

Analiza grafov skonstruiranih iz 3D struktur proteinov je tako pokazala, da bi lahko značilnosti takšnih grafov uporabili kot alternativni, oziroma dodaten validacijski korak pri reševanju struktur proteinov.

6 Bibliography

[1] A. Atilgan, P. Akan, and C. Baysal, Small-World Communication of Residues and Significance for Protein Dynamics. *Biophysical Journal* Vol. 86 (2004) 85-91. *(Cited on pages 13 and 14.)*

[2] G. Bagler and S. Sinha, Network properties of protein structures. *Physica A* 346 (2005) 27-33. *(Cited on page 14.)*

[3] B. Chakrabarty and N. Parekh, NAPS: Network Analysis of Protein Structures. *Nucleic Acids Research* Vol. 44, Web Server issue (2016) 375-382. *(Cited on page 11.)*

[4] W. Dhifli and A.B. Diallo, PGR: A Novel Graph Repository of Protein 3D-Structures. *J. Data Mining in Genomics & Proteomics* Vol. 6, No. 2 (2015) 172-175. *(Cited on page 10.)*

[5] L.H. Greene and V.A. Higman, Uncovering Network Systems Within Protein Structures. *J. Mol. Biol* 334 (2003) 781-791. *(Cited on pages 7, 14, and 21.)*

[6] L.H. Greene, Protein structure networks. *Briefings in Functional Genomics* Vol. 2, No. 6 (2012) 469-478. *(Not cited.)*

[7] l. Hong and J. Lei, *Scaling Law for the Radius of Gyration of Proteins and Its Dependence on Hydrophobicity*, `Published online in Wiley InterScience (www.interscience.wiley.com)`. (Viewed on: 20/1/2017.) *(Cited on page 14.)*

[8] G.J. Kleywegt, Validation of protein crystal structures. *Acta Crystallographica* D56 (2000) 249-265. *(Not cited.)*

[9] S. Peng and Y. Tsai, Adjusting protein graphs based on graph entropy. In *Proceedings of the 2013 International Conference on Intelligent Computing (ICIC 2013)*, 2014, Supplement 15. *(Cited on pages 20 and 21.)*

[10] K. Raman, N. Damaraju, and G.K. Joshi, The organisational structure of protein networks: revisiting the centrality-cethality hypothesis. *J. Syst Synth Biol* 8 (2014) 73-81. *(Not cited.)*

[11] G. Rhodes, *Crystallography Made Crystal Clear: A Guide for Users of Macromolecular Models*, Elsevier, Third Edition, 2006. *(Not cited.)*

[12] S. Vishveshwara and N. Kanan, Identification of Side-chain Clusters in Protein Structures by a Graph Spectral Method. *J. Mol. Biol* 292 (1999) 441-464. *(Not cited.)*

[13] S. Vishveshwara, K.V. Brinda, and N. Kanan, Protein structure: Insights from graph theory. *Journal of Theoretical and Computational Chemistry,* Vol. 1,No. 1 (2002) 000-000. *(Cited on page 3.)*

[14] D.B. West, *Introduction to Graph Theory*, Pearson Education, Second Edition, 2001. *(Not cited.)*

[15] H. Wu, Y. Zhang, W. Chen, and Z. Mu, Comparative analysis of protein primary sequences with graph energy. *Physica A* 437 (2015) 249-262. *(Cited on pages 19 and 20.)*

[16] *Protein structure*, `http://en.wikipedia.org/Protein_structure`. (Viewed on: 29/6/2017.) *(Cited on pages VII, 1, and 2.)*

[17] *Introduction to Protein Data Bank Format*, University of California, San Francisco. `https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html`. (Viewed on: 29/6/2017.) *(Cited on pages VII, 5, and 6.)*

[18] *NAPS: Network Analysis of Protein Structures*, `http://bioinf.iiit.ac.in/NAPS/`. (Viewed on: 25/1/2017.) *(Cited on page 11.)*

[19] *PGR: PROTEIN GRAPH REPOSITORY*, `http://wjdi.bioinfo.uqam.ca/`. (Viewed on: 25/1/2017.) *(Cited on page 10.)*

[20] *Dijkstra's Algorithm*, `https://en.wikipedia.org/wiki/Dijkstra27s_ algorithm`. (Viewed on: 29/6/2017.) *(Cited on page 13.)*

[21] *Perron-Frobenius theorem*, `https://en.wikipedia.org/wiki/Perron--Frobenius_theorem`. (Viewed on: 29/6/2017.) *(Cited on page 16.)*