5-2017

Quantifying the Structure of Misfolded Proteins Using Graph Theory

Walter G. Witt
East Tennessee State University

Recommended Citation

Witt, Walter G., "Quantifying the Structure of Misfolded Proteins Using Graph Theory" (2017). *Electronic Theses and Dissertations.* Paper 3244. https://dc.etsu.edu/etd/3244

Quantifying the Structure of Misfolded Proteins Using Graph Theory

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

by

Walter G. Witt

May 2017

Debra Knisley, Ph.D.

Jeff Knisley, Ph.D.

Robert Gardner, Ph.D.

Keywords: graph theory, single misfold, computational biology, spectral clustering

ABSTRACT

Quantifying the Structure of Misfolded Proteins Using Graph Theory

by

Walter G. Witt

The structure of a protein molecule is highly correlated to its function. Some diseases such as cystic fibrosis are the result of a change in the structure of a protein so that this change interferes or inhibits its function. Often these changes in structure are caused by a misfolding of the protein molecule. To assist computational biologists, there is a database of proteins together with their misfolded versions, called decoys, that can be used to test the accuracy of protein structure prediction algorithms. In our work we use a nested graph model to quantify a selected set of proteins that have two single misfold decoys. The graph theoretic model used is a three tiered nested graph. Measures based on the vertex weights are calculated and we compare the quantification of the proteins with their decoys. Our method is able to separate the misfolded proteins from the correctly folded proteins.

ACKNOWLEDGMENTS

A very special thanks is given my committee chair Dr. Debra Knisley for her vision and guidance throughout the project. Her inspiration and enthusiasm made the whole process extremely edifying. I would also like to thank committee member Dr. Jeff Knisley for developing the code used in previous research that motivated this project and his valuable insight throughout the research process. I want to give an additional thanks to committee member Dr. Robert Gardner for helping with the editing process of during the second draft.

Additional thanks are given to my family and friends for their support the past two years. I would also like to recognize Ram Samdrala and Michael Levitt for the creation of 'Decoys R Us' database that is used in the project. A final thank you is in order for the mathematics department at East Tennessee State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

In the area of discrete mathematics, there is a branch called graph theory, that researches the connection of entities. Graph theoretical models can be applied to numerous fields, which opens an interdisciplinary approach with a novel way to quantify complex networks. There is a great deal of work being done in computational biology to quantify protein structural characteristics and therefore identify and predict when a protein molecule will misfold. In the sequential sections we will improve upon a three tiered nested graph theoretical model that quantifies proteins.

1.1 Computational Biology

Biology, computer science, and mathematics all come together to form the field of computational biology. Molecular processes are very important to understanding life, and as a result, the research field bioinformatics was coined in 1970. It was brought into the Oxford Dictionary in 2002 [18]. Being able to understand and generate molecular information is crucial in computational biology.

Protein structure identification is an issue facing many computational biologists. Experiments are costly, time consuming, and meticulous, but with the addition of algorithms and well-defined mathematical models involving graph theory, this can be spearheaded in a cost effective efficient way [36].

Since much of the work in computational biology is focused on large data sets, it is difficult to apply current practices and algorithms to smaller sets [20]. Graph theoretical measures come into play here and provide method for quantifying protein structures. Through well-defined graph measures, it is possible to apply weights and quantify structures on several levels. This results in exponential data growth, given a small data source using a nested graph model.

Computational biologists provide a unique way to quantify the characteristics of proteins. The ability to quickly preform precise quantification at a reduced cost may catapult this field in the limelight.

1.2 Motivation

Knisley, Knisley and Herron (KKH) introduced partitioning a protein into domains and using those domains with weight measures of the amino acids to create a weighted top-level graph. The focus of KKH was on the single mutation point of amino acids chain in the cystic fibrosis membrane conductance regulator (CFTR) protein [20]. KKH partitioned nucleotide binding domain one (NBD1) into 8 subsequences of 20 amino acids guided from the secondary structure of the protein and then created a nest graph model to represent the NBD1 of CFTR.

This process allows the invariants to be controlled through well-defined graph measures. The contact graph is created from a sequence of amino acids, which represent the vertices, and edges, that are determined by the proximity measure of 8 angstroms [20]. The middle level, where the amino acid subsequences are located, defines the top level of the network as subdomains. These subdomains represent vertices of the top level. The top level is assigned weights based on its corresponding amino acid descriptors from the middle level.

The process allows back tracking to define new invariants. This is where after quantification, it is possible to analyze the levels with different weighted measures to improve the quantification process.

1.3 Process to be Improved and Compared

How were the invariants chosen before? Is there a way to choose new domains that are well-defined with a set of data? These questions started the process of trying to improve the methods of (KKH).

(KKH) used a partitioning of the subsequences for the domains that were constructed by the guideline of having only one type of secondary structure [20] . This yields domains that have different subsequence lengths. The original contact graph used a cutoff value of 8 angstroms during its construction.

In the new process, the contact graph being used will have a cutoff value of 7 angstroms per Silveira et al., which stated that at 7 angstroms, all connections between amino acids are concise [13]. Keeping in line with having different subsequence lengths, I introduce spectral clustering as a partitioning method. Due to being a well-defined clustering algorithm, it is believed that spectral clustering will be able to partition the proteins from the Protein Data Bank (PDB) and Decoys R Us into domains that are highly connected. Once the subdomains are representing the top level as vertices, there will be a floor of at least 3 edges between subdomains needed to create an edge between respective nodes at the top level. This allows well-defined graph theoretic measures to quantify these domains in a way that provides additional insight in the proteins in question.

The proteins being compared are 2cro, 2ci2 and 1sn3. Each of these three proteins have 2 decoys from the database Decoys R Us. Note 1sn3 is an outdated protein, and is superseded by 2sn3 [37]. We continue with the use of 1sn3 in this model showing both scenarios where 1sn3 and its decoys are present and another when they're not.

2 BACKGROUND

Biology, graph theory, data, and algorithms are the building blocks of this thesis. Essential information that is going to be used later in Section 3, Process, is discussed in this section. Special attention to definitions and measures is crucial for understanding the process implemented, as Section 3 becomes complex rather quickly. The building blocks are located in biology.

With the introduction of computational biology, we can see the interaction between graph theory and life. These classical models are given purpose with their representation of biomolecules known as proteins.

2.1 Biology Interdisciplinary

The structure of biomolecules are long chains of amino acids which are called proteins. A protein is a polypeptide, which is a distinct sequence of amino acids. All amino acids have the same backbone, notice the non-shaded part in Figure 1. This is the backbone. The distinction of amino acids come from their R group, which is the shaded part of Figure 1. Amino acids are represented by letters of the alphabet.

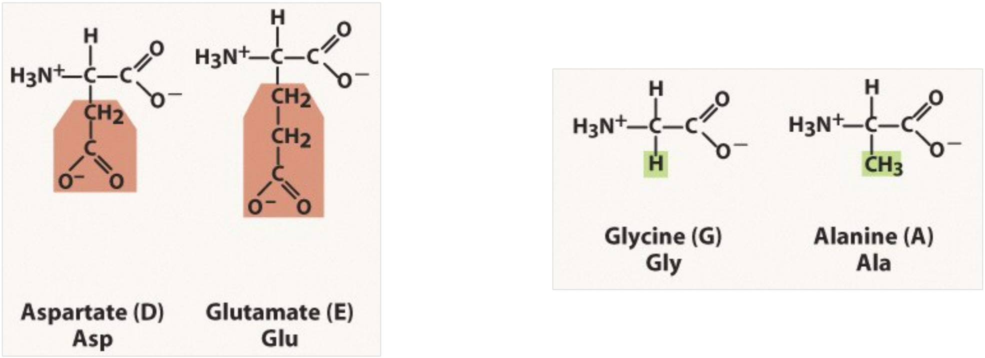


Figure 1: Four amino acids with backbone (unshaded) and R group (shaded)

Table 1: Amino Acids and their corresponding letters.

Amino Acid	Letter
alanine	A
arginine	R
asparagine	N
aspartic Acid	D
cysteine	C
glutamine	Q
glutamic acid	E
glycine	G
histidine	H
isoleucine	I
leucine	L
lysine	K
methionine	M
phenylalanine	F
proline	P
serine	S
threonine	T
tryptophan	W
tyrosine	Y
valine	V

The primary structure of a protein is composed of a chain of amino acids. The uniqueness of the chain determines the characteristics of the protein and it's resulting structure. Once the chain is completed, the primary structure folds upon itself creating a secondary structure. The structures are recorded in a database that are constantly improved through scientific research.

The Protein Data Bank (PDB) is the primary data repository for protein and DNA three dimensional structures. A PDB file can be extracted from the repository that contains hierarchical structure regarding atom names and coordinates for said protein [12].

These protein sequences are being studied intensively. Investigation of these relationships between the protein protein interaction (PPI) and sequence formation enhances understanding of how proteins function. The cognition of this will yield: protein folding, prediction of protein structures, patterns of molecular evolution, protein engineering, and drug design. Researchers focus on these computations in hopes of providing insights into the workings of complex biological systems [36, 12].

2.2 Decoys R Us

Decoys R Us is a database of computer generated protein structures that have been used in the computational biology community since 2000. These highly respected models aid in the development of current knowledge of protein structures [29]. The purpose of the database is to improve prediction scoring methods by providing an alternative decoy database to measure against a known database of protein structures.

Decoys R Us includes a database of single misfold proteins. It is comprised of 26 incorrect proteins based upon 23 correct protein structures.

2.3 Graph Theory

In discrete mathematics, there is a field of graph theory. This is the study of two disjoint sets, elements and relations. For example, picture computers that represent elements and the Internet which represents the relation of the computers. The computers are elements that are connected through the Internet. The following is extracted from several sources [17, 13, 3, 25, 16, 5].

A **graph** $G = (V, E)$ is an ordered pair that has disjoint sets V and E. The elements of V are called vertices or nodes, and the elements of E are called edges. Each edge has a set of one or two vertices assoiciated to it, which are called its endpoints.

The **adjacency matrix** of graph G denoted A_G, is the matrix whose rows and columns are both indexed by identical ordering of V_G, such that

$$A_G[u, v] = \begin{cases} 1 \text{ if } u, v \in E \\ \\ 0 \text{ otherwise} \end{cases}$$

The **degree matrix** of a graph G denoted D_G, is a diagonal $n \times n$ matrix for which

$$d_{u,v} = \begin{cases} \text{degree of } G_u \text{ if } u = v \\ \\ 0 \text{ otherwise} \end{cases}$$

The **Laplacian matrix** of a graph G, denoted L_G is $L_G = D_G - A_G$

A **closed walk** of a graph G, is a sequence of pairwise adjacent vertices beginning and ending with the same vertex. The **trace** (diagonal) of the A_G^k, where k is the number of walks desired. The trace of a power of the adjacency matrix is a method for counting the number of closed walks.

The following are node measures are from networkX's algorithm library, it can be found in [25]. NetworkX is a library which specializes in graph theoretic measure algorthims.

The **eccentricity** of a node v in graph G is the maximum distance from v to all other nodes in G.

The **node clique number** returns the largest maximal clique containing each node given. The formal definition follows, a subset S of V_G is called a **clique** if every pair of vertices in S is joined by at least one edge, and no proper superset of S has this property. So a clique on G is the maximum subset of mutually adjacent vertices in G [17].

The **degree centrality** of a node v in the graph G is the fraction of nodes it is connected to. The degree centrality values are normalized by dividing by the maximum possible degree in a simple graph, which is $n-1$ where n is the number of nodes in G.

The **closeness centrality** of a node u in graph G is the reciprocal of the sum of the shortest path distances from u to all $n-1$ other nodes [16]. This is normalized by the sum of possible distances $n-1$,

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

where $d(v,u)$ is the shortest path between v and u and n is the number of nodes in graph G.

The **betweenness centrality** of a node v is the sum of the fraction of all pairs of shortest paths that pass through v

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

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t) paths, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s, t. If $s = t, \sigma(s,t) = 1$, and if $v \in s, t, \sigma(s,t|v) = 0$ [6].

The **current flow closeness centrality** is a variant of closeness based on effective resistance between nodes in a network.

$$C_{CC}(s) = \frac{n}{\sum_{s \neq t} p_{st}(s) - p_{st}(t)} \text{ for all } s \in V,$$

where $p_{st}(s) - p_{st}(t)$ is the effective resistance. Another name for this algorithm is information centrality [7].

The **current flow betweenness centrality** of nodes is an electric current model for information spreading, which is defined by

$$C_{CB}(v) = \frac{1}{n_B} \sum_{s,t \in V} \tau_{st}(v) \text{ for all } v \in V,$$

where $n_B = (n-1)(n-2)$ [7].

The **eigenvector centrality** is the calculation of the centrality for a node based on the centrality of its neighbors. The eigenvector centrality for node i is x_i. $Ax = \lambda x$, where A is the adjacency matrix of graph G with eigenvalue λ.

The **communicability centrality**, also known as the subgraph centrality, of a node n is the sum of closed walks of all lengths starting and ending at node n. Communicability centrality of a node u of graph G can be found by the spectral decomposition of the adjacency matrix. It is defined as

$$SC(i) = \sum_{j=1}^{N} \left(v_j^i\right)^2 e^{\lambda_j},$$

where v_j^i is the $i - th$ component of the orthonormal basis R^N composed by the eigenvectors A associated to the eigenvalues λ_j [14].

2.4 Spectral Clustering

Clustering algorithms are used for data that is highly connected [24]. Spectral clustering is used for data that is connected, but not necessarily isolated in a way that convex optimization can take place. The basic goal is to divide the data points of a given graph into clusters of similar points that are different from other clusters [23]. This yields a specified number of clusters of points that are mathematically similar.

In order to implement spectral clustering, the number of clusters need to be determined. The number of clusters is directly related to the lowest values corresponding to the eigenvalues of the normalized Laplacian of the graph theoretic model [24, 23].

The graph theoretic model needs to be represented as an adjacency matrix A. The degree matrix D is then used to create a Laplacian matrix,

$$L = D - A.$$

Then the normalized Laplacian is constructed from L,

$$L_n = D^{-1/2} L D^{-1/2}$$

and the eigenvalues are found for the normalized Laplacian. The eigenvalues are then plotted and the total number n of the lowest values are picked off to form the number of clusters for the spectral clustering algorithm [24, 23].

When choosing the number of clusters it is important to plot the eigenvalues of the normalized Laplacian in order from least to greatest. The goal is to pick a gap between the eigenvalues. This gap will represent the optimal amount of clusters for the algorithm to compute. It also important to note that if several gaps are shown, the cluster value chosen should be reasonable. It would not make sense to break a network with 60 nodes into 30 clusters.

The algorithm that will be implemented is taken from scikit-learn. The only parameter being used is the number of clusters.

2.5 Amino Acid Descriptors

Every amino acid is represented as a molecule, as seen in Figure 1. These molecules have key characteristics that make each amino acid unique. The R group is what gives these amino acids their distinction.

Graph theory can represent another characteristic of these amino acids. When these molecules are represented using chemical bonds, it can be shown using molecular topology that these amino acids can be represented by graph theoretical models, where the atoms are the nodes and the chemical bonds are the edges of the graph model [34].

These two representations give us a plethora of descriptors to aid in the quantification of all 20 amino acids. Using the AAindex, which is a database of numerical indices representing physiochemical and biochemical properties [1], and graph thoeretic measures on molecular topologies, a table of descriptors is compiled for each amino acid.

Although there are numerous descriptors available, there are 21 in the table used in our model. Of the 21 used, only 9 are taken to quantify the proteins in question. The following is a description of the 9 descriptors used [10, 9, 1].

The 9 descriptors used are defined as follow:

The **domination number** of a graph G, is the cardinality of a minimum set S of vertices such that every vertex of G is either in S or a neighbor of a vertex in S [17]. The maximum domination number is the maximum instead of the minimum.

chargedonar is the parameter of charge transfer donor capability [9].

coilconformation is the Chou-Fasman parameter of the coil conformation [9].

chargetransfr is the parameter of charge transfer capability [9].

Balaban is the Balaban Index, which is

$$J = \frac{m}{\gamma + 1} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(S_i S_j\right)^{-1/2}$$

where n are the nodes and m are the edges of the molecular graph. $\gamma = m - n + 1$ is the cyclomatic number, which the smallest number of edges that need to be removed in order so that no graph cycle is remaining, and $S_{i,j}$ are the sum of entries in the respective rows of the graph distance matrix [2].

EIIP is the electron-ion interaction potential [8].

Plr is the side-chain polarity. If a side chain is reactive with water, then it is said to be polar [11]. A value of 1 is assigned to the amino acid if it is polar and a value of 0 is assigned if it is not polar.

The **circumference** of a graph G, represented as c, is the length of the longest cycle. If no cycle is present the value 0 will recorded [21].

Table 2: Amino Acids Descriptor Index.

Descriptor Index	Name
G	maximal domination number
chargedonar	parameter of charge transfer donor capability
coilconformation	Chou-Fasman parameter of the coil conformation
chargetransfr	parameter of charge transfer capability
Balaban	The Balaban Index
EIIP	electron-ion interaction potential
Plr	side-chain polarity
c	circumference of the molecular topology
averagehydrophcity	normalized average hydrophobicity scales

2.6 Data Analysis

When there is data present, sometimes it needs to preprocessed so we can see what story the data is trying to tell us. To find out the story, we use l_2 normalization of preprocessing from the scikit-learn, as well as, the dendrogram function from the same package.

A **Norm** must follow these four axioms:

Nonnegativity: $||x|| \geq 0$

positivity: $||x|| = 0$ iff $x = 0$

Homogeneity: $||cx|| = |c|||x||$

Triangle Inequality: $||x + y|| \leq ||x|| + ||y||$

These axioms make sure that any non-zero vector can be normalized. The result is a normed linear space, written of the form $|| \cdot ||$ [19].

The l_2 norm, which is called the Euclidean norm of a vector $x = [x_1 \dots x_n]^T \in \mathbb{C}^n$ is as follows:

$$||x||_2 = \left(|x_1|^2 + \cdots + |x_n|^2\right)^{1/2}$$

A **Dendrogram** is the application of hierarchical clustering. It is a tree-type diagram showing a series of steps that groups information into clusters. This method of clustering uses single linkage clustering to distinguish between steps taken [15].

Single-linkage clustering is a method of clustering analysis where distance between clusters is defined to be the least distance between the pair, where one of the data points is in the group or cluster [15].

3 IMPLEMENTATION

This section describes the technical process, in python, of quantifying protein structures. The grouping for the domains of the mid level graph is a spectral clustering algorithm implemented from the scikit-learn package [31], and the graph measures used to define weights for the top level graph are implemented from the networkX package [25] in python. After quantification a table is compiled using the pandas package [27] and subsequently analyzed with a hierarchical clustering algorithm in the SciPy package [32] to create a dendrogram. The dendrogram is a measure of how well the invariants chosen work for our quantification.

3.1 Process

The quantification process starts with a Protein Data Bank (PDB) file. In this section 2ci2, a A PDB file is uploaded and its sequence (chain) of amino acids are listed. The ID numbers for the corresponding amino acids are defined, as proteins are not all the same length. This is used to create a contact graph in networkX [25].

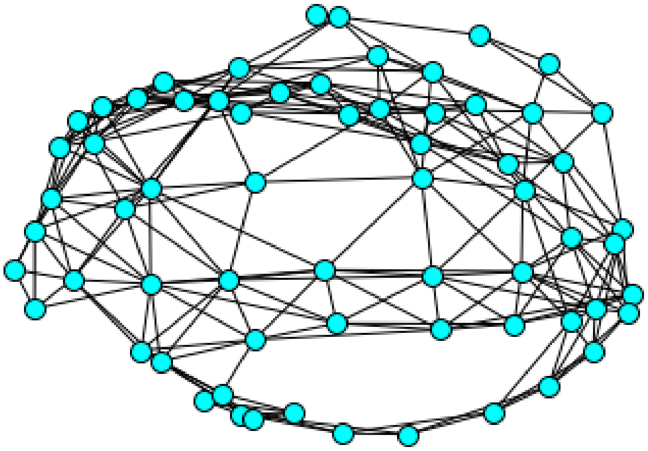
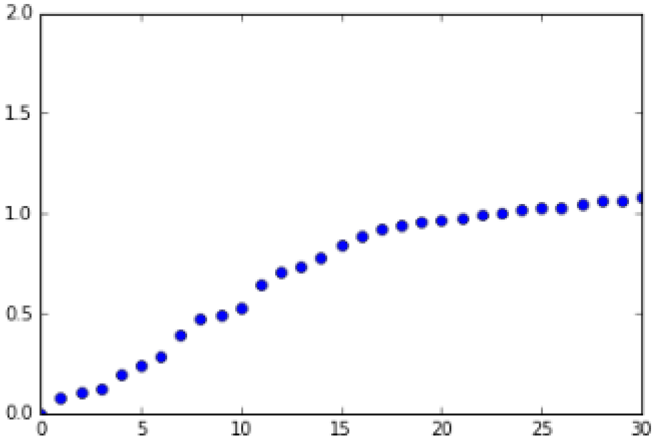


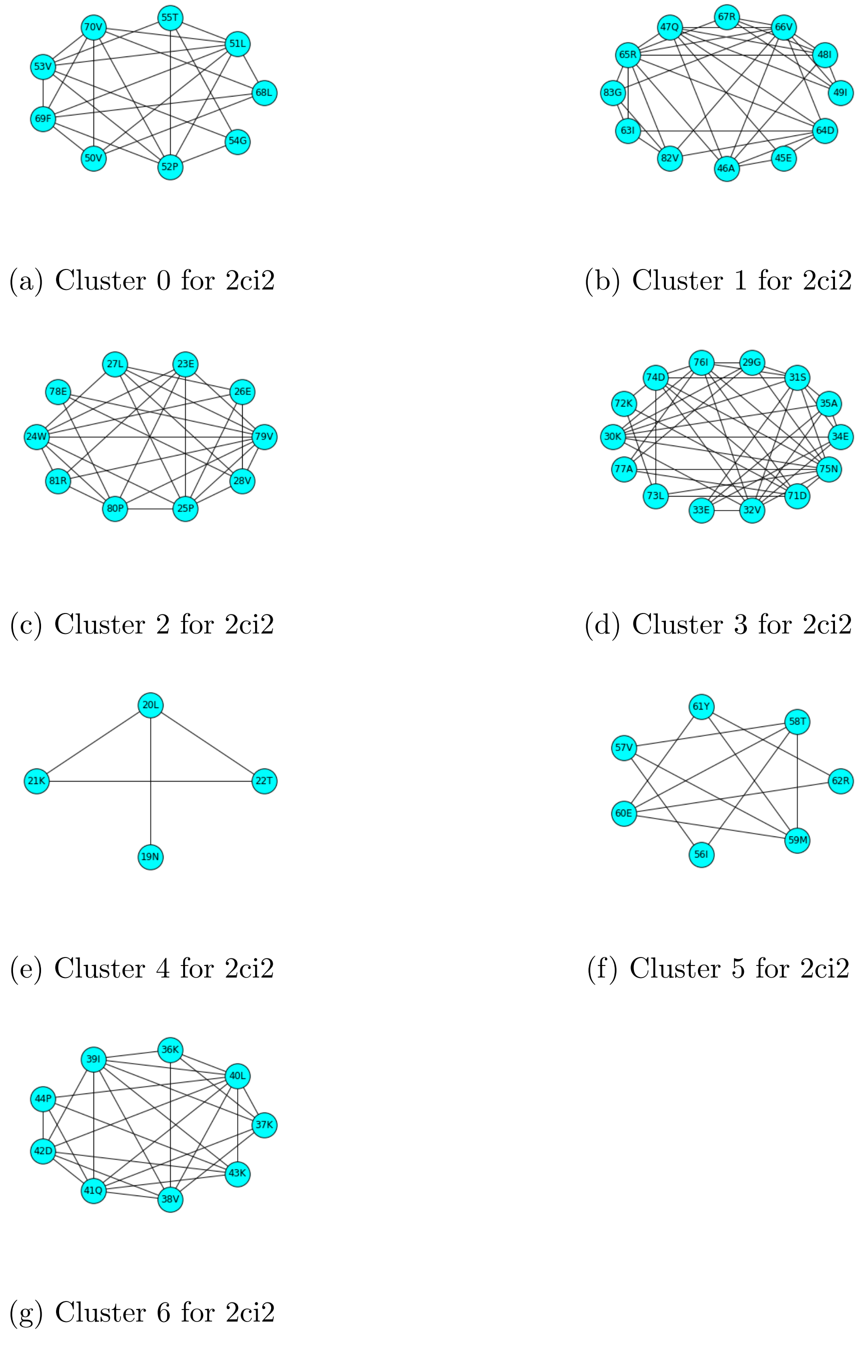
Figure 2: Contact Graph for 2ci2

The graph theoretic model of the protein shows high connectivity. In order to obtain a nice number of domains for the mid level graph, a normalized Laplacian is constructed through the networkX package. After the normalized Laplacian is constructed, the eigenvalue problem is solved and the lowest values will be selected.

Figure 3: Eigenvalues of the normalized Laplacian



The number of clusters is 7 based upon the lowest eigenvalues. The number of clusters for each protein is determined for the true proteins and the cluster number will be applied to the respective decoys. That is, the cluster number for 2ci2 is 7; thus its respective decoys will be partitioned into 7 clusters as well. The adjacency matrix is then constructed using the pandas package. The spectral clustering algorithm from the scikit-learn package is implemented. Spectral clustering provides a method to account for the connectivity of the nodes in the adjacency matrix. The data is merged and the nodes from the adjacency matrix are now labeled to the corresponding cluster produced from the spectral clustering algorithm. The groups in Figure 4 are shown to contain different clusters of amino acids. This is the purpose of using spectral clustering. It was meant to group the protein in ways that take into account its connectivity.



(a) Cluster 0 for 2ci2

(b) Cluster 1 for 2ci2

(c) Cluster 2 for 2ci2

(d) Cluster 3 for 2ci2

(e) Cluster 4 for 2ci2

(f) Cluster 5 for 2ci2

(g) Cluster 6 for 2ci2

Figure 4: Clusters for protein 2ci2

The top level graph is constructed with nodes that correspond to the respective cluster in mid level. These nodes are labeled $\{0, 1, 2 \ldots, n-1\}$ where n is the number of clusters used in the spectral clustering algorithm. The method for determining if clusters are connected in the top level graph is as follows: if there are 3 or more edges connecting clusters, then an edge is used to connect the nodes in the top level graph. The top level graph for 2ci2 is shown in Figure 5.

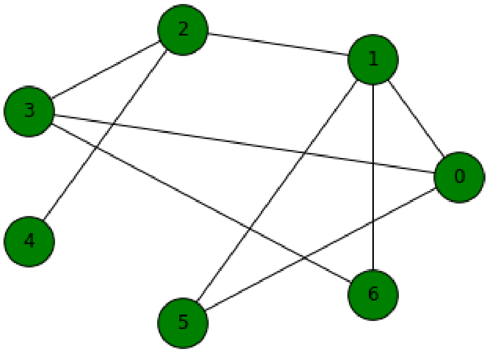
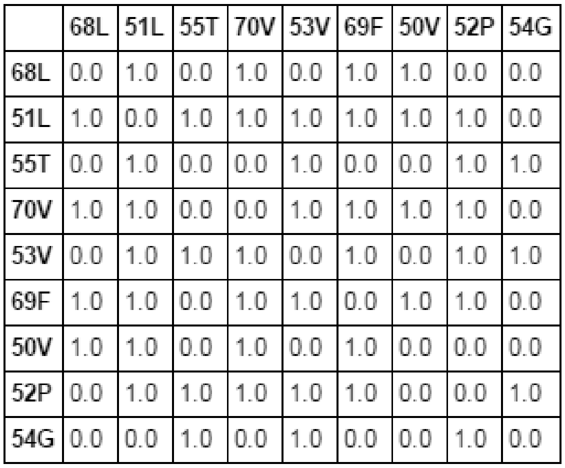


Figure 5: Top level graph from respective clusters

3.2 Quantification

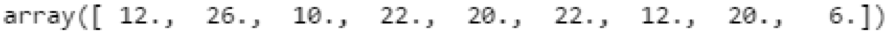
After the top level graph is created, the nodes representing the respective clusters are given weights. This process starts with the adjacency matrix for each cluster in Figure 4 that represents the corresponding node in Figure 5.

Table 3: 2ci2 Cluster 0 Adjacency Matrix



	68L	51L	55T	70V	53V	69F	50V	52P	54G
68L	0.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
51L	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
55T	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0
70V	1.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0
53V	0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0
69F	1.0	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0
50V	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
52P	0.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0
54G	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0

The measure being used is the number of complete walks of length 3. This is calculated by cubing the adjacency matrix and calculating the trace of the cubed adjacency matrix.



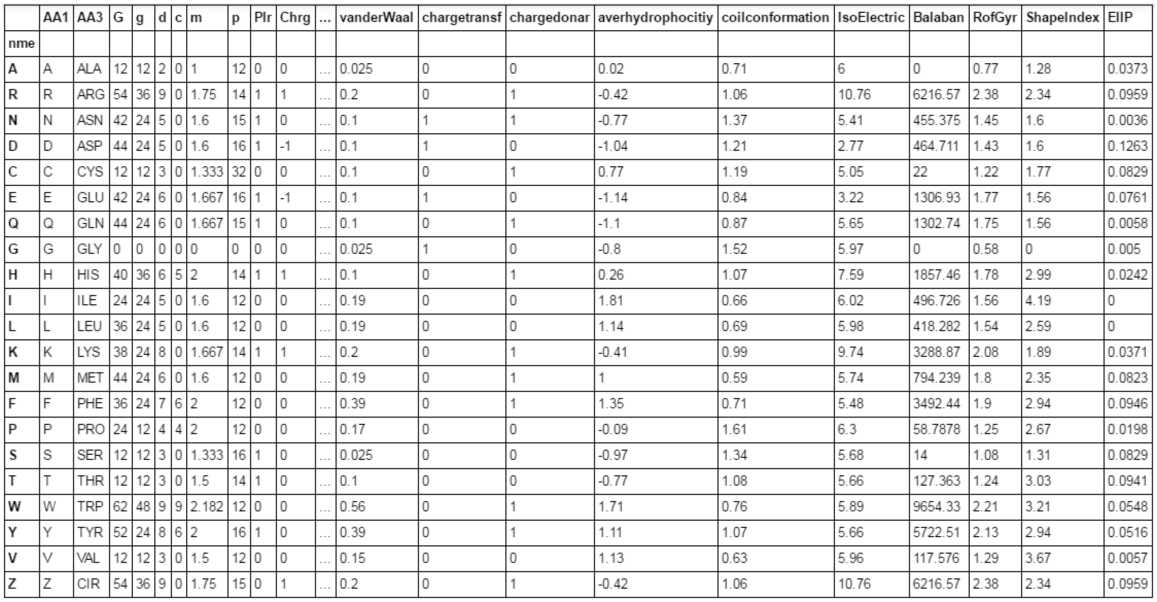
```
array([ 12.,  26.,  10.,  22.,  20.,  22.,  12.,  20.,   6.])
```

Figure 6: 2ci2 Cluster 0 Complete Walks of 3

Coefficients in the resulting array are assigned as weights, respectively, to the corresponding nodes.

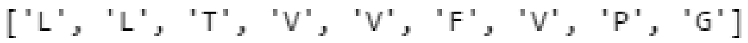
The amino acid descriptors (AAD) data frame, Table 4, which shows quantification for every amino acid used in this model. These quantities are descriptors taken from molecular topology, graph theoretic measures and the amino acid index. Note that not every value recorded in the table is used for the final quantification process. The descriptors defined in the previous section are the ones that are used.

Table 4: Amino acids descriptors



	AA1	AA3	G	g	d	c	m	p	Plr	Chrg	...	vanderWaal	chargetransf	chargedonar	averhydrophocitiy	coilconformation	IsoElectric	Balaban	RofGyr	ShapeIndex	EIIP
nme																					
A	A	ALA	12	12	2	0	1	12	0	0	...	0.025	0	0	0.02	0.71	6	0	0.77	1.28	0.0373
R	R	ARG	54	36	9	0	1.75	14	1	1	...	0.2	0	1	-0.42	1.06	10.76	6216.57	2.38	2.34	0.0959
N	N	ASN	42	24	5	0	1.6	15	1	0	...	0.1	1	1	-0.77	1.37	5.41	455.375	1.45	1.6	0.0036
D	D	ASP	44	24	5	0	1.6	16	1	-1	...	0.1	1	0	-1.04	1.21	2.77	464.711	1.43	1.6	0.1263
C	C	CYS	12	12	3	0	1.333	32	0	0	...	0.1	0	1	0.77	1.19	5.05	22	1.22	1.77	0.0829
E	E	GLU	42	24	6	0	1.667	16	1	-1	...	0.1	1	0	-1.14	0.84	3.22	1306.93	1.77	1.56	0.0761
Q	Q	GLN	44	24	6	0	1.667	15	1	0	...	0.1	0	1	-1.1	0.87	5.65	1302.74	1.75	1.56	0.0058
G	G	GLY	0	0	0	0	0	0	0	0	...	0.025	1	0	-0.8	1.52	5.97	0	0.58	0	0.005
H	H	HIS	40	36	6	5	2	14	1	1	...	0.1	0	1	0.26	1.07	7.59	1857.46	1.78	2.99	0.0242
I	I	ILE	24	24	5	0	1.6	12	0	0	...	0.19	0	0	1.81	0.66	6.02	496.726	1.56	4.19	0
L	L	LEU	36	24	5	0	1.6	12	0	0	...	0.19	0	0	1.14	0.69	5.98	418.282	1.54	2.59	0
K	K	LYS	38	24	8	0	1.667	14	1	1	...	0.2	0	1	-0.41	0.99	9.74	3288.87	2.08	1.89	0.0371
M	M	MET	44	24	6	0	1.6	12	0	0	...	0.19	0	1	1	0.59	5.74	794.239	1.8	2.35	0.0823
F	F	PHE	36	24	7	6	2	12	0	0	...	0.39	0	1	1.35	0.71	5.48	3492.44	1.9	2.94	0.0946
P	P	PRO	24	12	4	4	2	12	0	0	...	0.17	0	0	-0.09	1.61	6.3	58.7878	1.25	2.67	0.0198
S	S	SER	12	12	3	0	1.333	16	1	0	...	0.025	0	0	-0.97	1.34	5.68	14	1.08	1.31	0.0829
T	T	THR	12	12	3	0	1.5	14	1	0	...	0.1	0	0	-0.77	1.08	5.66	127.363	1.24	3.03	0.0941
W	W	TRP	62	48	9	9	2.182	12	0	0	...	0.56	0	1	1.71	0.76	5.89	9654.33	2.21	3.21	0.0548
Y	Y	TYR	52	24	8	6	2	16	1	0	...	0.39	0	1	1.11	1.07	5.66	5722.51	2.13	2.94	0.0516
V	V	VAL	12	12	3	0	1.5	12	0	0	...	0.15	0	0	1.13	0.63	5.96	117.576	1.29	3.67	0.0057
Z	Z	CIR	54	36	9	0	1.75	15	0	1	...	0.2	0	1	-0.42	1.06	10.76	6216.57	2.38	2.34	0.0959

The (AAD) values in Table 4 are now associated with the the amino acid labels of Figure 7.

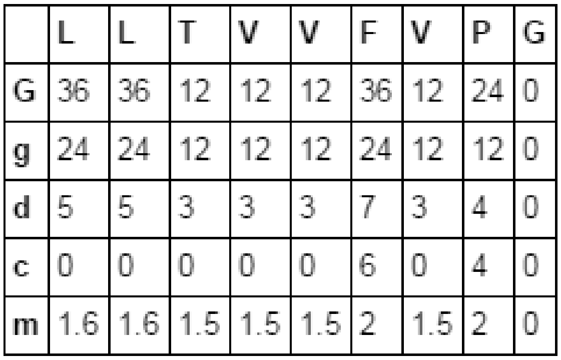


```
['L', 'L', 'T', 'V', 'V', 'F', 'V', 'P', 'G']
```

Figure 7: Amino acids being used in cluster 0

The (AAD) are now concatenated to create a new data frame. This is shown in Table 5.

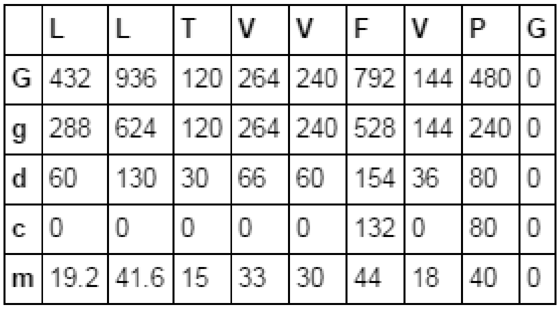
Table 5: Concatenated amino acids with first 5 (shown) descriptors for cluster 0



	L	L	T	V	V	F	V	P	G
G	36	36	12	12	12	36	12	24	0
g	24	24	12	12	12	24	12	12	0
d	5	5	3	3	3	7	3	4	0
c	0	0	0	0	0	6	0	4	0
m	1.6	1.6	1.5	1.5	1.5	2	1.5	2	0

Next the array of counts of closed walks of length 3 for cluster 0 in Figure 7 are then multiplied, as scalars, onto the the concatenated matrix in Table 5. The result of this process is represented in Table 6.

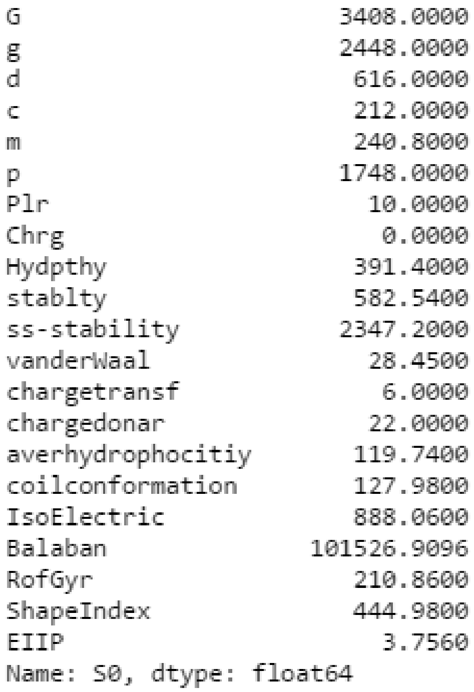
Table 6: Cluster 0 with first 5 (shown) descriptors with weights applied



	L	L	T	V	V	F	V	P	G
G	432	936	120	264	240	792	144	480	0
g	288	624	120	264	240	528	144	240	0
d	60	130	30	66	60	154	36	80	0
c	0	0	0	0	0	132	0	80	0
m	19.2	41.6	15	33	30	44	18	40	0

A linear combination for each of the descriptor value is taken. These linear combinations will serve as weights for each respective node in the top level graph. The result for cluster 0 of 2ci2 is shown Table 7.

Table 7: Linear combination for each weighted descriptor in cluster 0

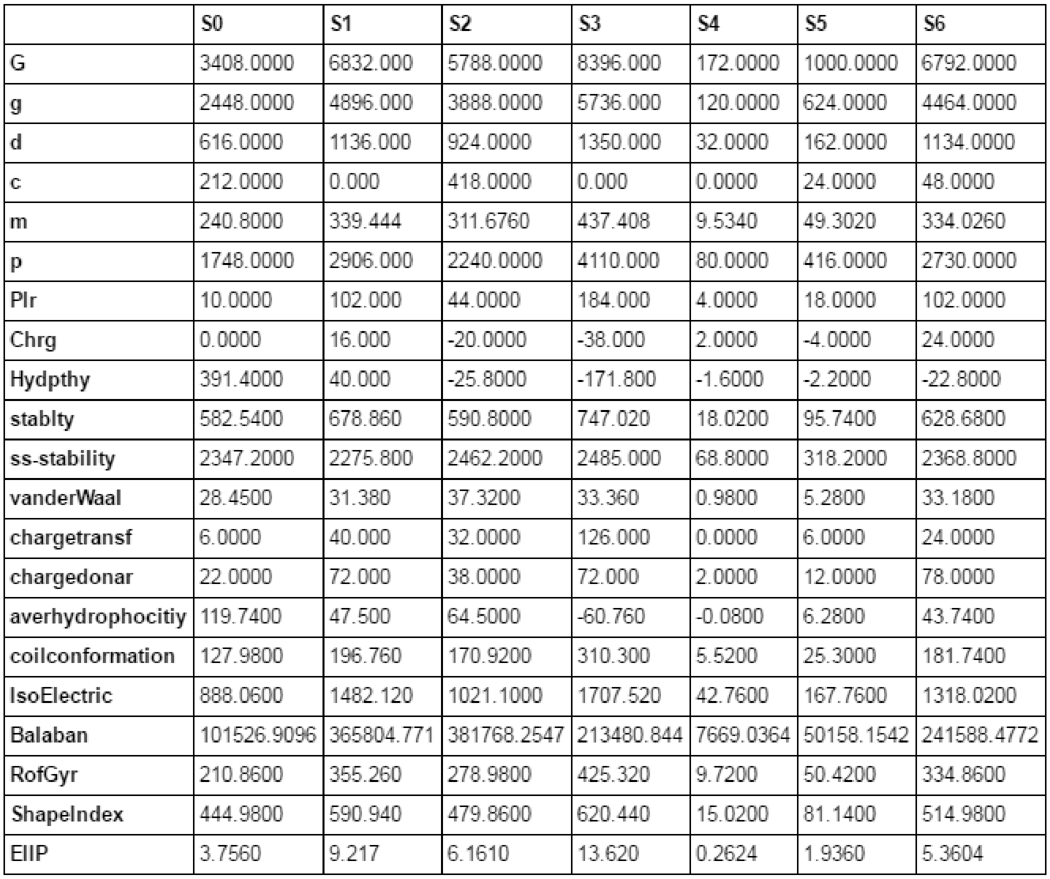


```
G                        3408.0000
g                        2448.0000
d                         616.0000
c                         212.0000
m                         240.8000
p                        1748.0000
Plr                        10.0000
Chrg                        0.0000
Hydpthy                   391.4000
stablty                   582.5400
ss-stability             2347.2000
vanderWaal                 28.4500
chargetransf                6.0000
chargedonar                22.0000
averhydrophocitiy         119.7400
coilconformation          127.9800
IsoElectric               888.0600
Balaban                101526.9096
RofGyr                    210.8600
ShapeIndex                444.9800
EIIP                        3.7560
Name: S0, dtype: float64
```

After this process is repeated for all the clusters, the respective nodes in the top level graph are represented by the calculated weights. All the linear combinations of the descriptor weights for each cluster are concatenated into a new data frame. This can be seen in Table 8, where $S_0, S_1, \ldots, S_{n-1}$, are the corresponding clusters.

Now that we have weights for the top level graph, we implement graph theoretic measures for the top level graph. The measures defined in Section 2.3 are used on the top level graph. These measure are specifically chosen due to the fact that they assign a value for each node represented in the top level graph. Now that each node in the top level graph can be measured, we can apply the measures to the descriptor weights for the top level graph in Table 8.

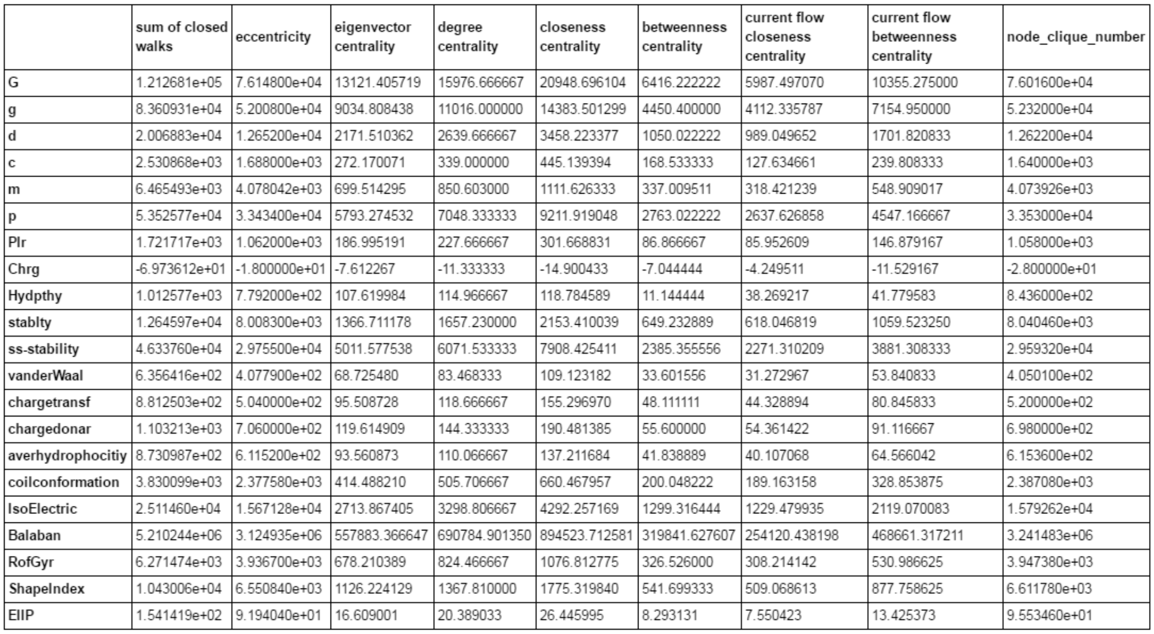
Table 8: Descriptor weights for the top level graph



	S0	S1	S2	S3	S4	S5	S6
G	3408.0000	6832.000	5788.0000	8396.000	172.0000	1000.0000	6792.0000
g	2448.0000	4896.000	3888.0000	5736.000	120.0000	624.0000	4464.0000
d	616.0000	1136.000	924.0000	1350.000	32.0000	162.0000	1134.0000
c	212.0000	0.000	418.0000	0.000	0.0000	24.0000	48.0000
m	240.8000	339.444	311.6760	437.408	9.5340	49.3020	334.0260
p	1748.0000	2906.000	2240.0000	4110.000	80.0000	416.0000	2730.0000
Plr	10.0000	102.000	44.0000	184.000	4.0000	18.0000	102.0000
Chrg	0.0000	16.000	-20.0000	-38.000	2.0000	-4.0000	24.0000
Hydpthy	391.4000	40.000	-25.8000	-171.800	-1.6000	-2.2000	-22.8000
stablty	582.5400	678.860	590.8000	747.020	18.0200	95.7400	628.6800
ss-stability	2347.2000	2275.800	2462.2000	2485.000	68.8000	318.2000	2368.8000
vanderWaal	28.4500	31.380	37.3200	33.360	0.9800	5.2800	33.1800
chargetransf	6.0000	40.000	32.0000	126.000	0.0000	6.0000	24.0000
chargedonar	22.0000	72.000	38.0000	72.000	2.0000	12.0000	78.0000
averhydrophocitiy	119.7400	47.500	64.5000	-60.760	-0.0800	6.2800	43.7400
coilconformation	127.9800	196.760	170.9200	310.300	5.5200	25.3000	181.7400
IsoElectric	888.0600	1482.120	1021.1000	1707.520	42.7600	167.7600	1318.0200
Balaban	101526.9096	365804.771	381768.2547	213480.844	7669.0364	50158.1542	241588.4772
RofGyr	210.8600	355.260	278.9800	425.320	9.7200	50.4200	334.8600
ShapeIndex	444.9800	590.940	479.8600	620.440	15.0200	81.1400	514.9800
EIIP	3.7560	9.217	6.1610	13.620	0.2624	1.9360	5.3604

This is the same process as before, as in the graph theoretic measures in the top level graph for each node are represented as scalars and multiplied through their corresponding clusters $S_0, S_1, \ldots, S_{n-1}$. After this is done, we take a linear combination of each measure for every amino acid descriptor. The resulting information is concatenated into a spreadsheet for the corresponding protein. The results for protein 2ci2 can be seen in Table 9.

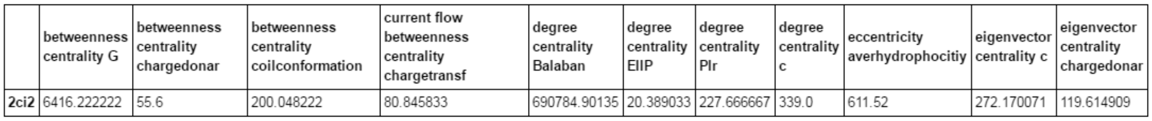
Table 9: 2ci2 top level graph measures with amino acid descriptors



	sum of closed walks	eccentricity	eigenvector centrality	degree centrality	closeness centrality	betweenness centrality	current flow closeness centrality	current flow betweenness centrality	node_clique_number
G	1.212681e+05	7.614800e+04	13121.405719	15976.666667	20948.696104	6416.222222	5987.497070	10355.275000	7.601600e+04
g	8.360931e+04	5.200800e+04	9034.808438	11016.000000	14383.501299	4450.400000	4112.335787	7154.950000	5.232000e+04
d	2.006883e+04	1.265200e+04	2171.510362	2639.666667	3458.223377	1050.022222	989.049652	1701.820833	1.262200e+04
c	2.530868e+03	1.688000e+03	272.170071	339.000000	445.139394	168.533333	127.634661	239.808333	1.640000e+03
m	6.465493e+03	4.078042e+03	699.514295	850.603000	1111.626333	337.009511	318.421239	548.909017	4.073926e+03
p	5.352577e+04	3.343400e+04	5793.274532	7048.333333	9211.919048	2763.022222	2637.626858	4547.166667	3.353000e+04
Plr	1.721717e+03	1.062000e+03	186.995191	227.666667	301.668831	86.866667	85.952609	146.879167	1.058000e+03
Chrg	-6.973612e+01	-1.800000e+01	-7.612267	-11.333333	-14.900433	-7.044444	-4.249511	-11.529167	-2.800000e+01
Hydpthy	1.012577e+03	7.792000e+02	107.619984	114.966667	118.784589	11.144444	38.269217	41.779583	8.436000e+02
stablty	1.264597e+04	8.008300e+03	1366.711178	1657.230000	2153.410039	649.232889	618.046819	1059.523250	8.040460e+03
ss-stability	4.633760e+04	2.975500e+04	5011.577538	6071.533333	7908.425411	2385.355556	2271.310209	3881.308333	2.959320e+04
vanderWaal	6.356416e+02	4.077900e+02	68.725480	83.468333	109.123182	33.601556	31.272967	53.840833	4.050100e+02
chargetransf	8.812503e+02	5.040000e+02	95.508728	118.666667	155.296970	48.111111	44.328894	80.845833	5.200000e+02
chargedonar	1.103213e+03	7.060000e+02	119.614909	144.333333	190.481385	55.600000	54.361422	91.116667	6.980000e+02
averhydrophocitiy	8.730987e+02	6.115200e+02	93.560873	110.066667	137.211684	41.838889	40.107068	64.566042	6.153600e+02
coilconformation	3.830099e+03	2.377580e+03	414.488210	505.706667	660.467957	200.048222	189.163158	328.853875	2.387080e+03
IsoElectric	2.511460e+04	1.567128e+04	2713.867405	3298.806667	4292.257169	1299.316444	1229.479935	2119.070083	1.579262e+04
Balaban	5.210244e+06	3.124935e+06	557883.366647	690784.901350	894523.712581	319841.627607	254120.438198	468661.317211	3.241483e+06
RofGyr	6.271474e+03	3.936700e+03	678.210389	824.466667	1076.812775	326.526000	308.214142	530.986625	3.947380e+03
ShapeIndex	1.043006e+04	6.550840e+03	1126.224129	1367.810000	1775.319840	541.699333	509.068613	877.758625	6.611780e+03
EIIP	1.541419e+02	9.194040e+01	16.609001	20.389033	26.445995	8.293131	7.550423	13.425373	9.553460e+01

After the top level graph measures with amino acid descriptors table is created, we select a subset of entries to compare for the proteins in question. The graph measure is listed first with the descriptor following, as seen in Table 10. It is listed as the column header, while the protein quantified represents the row.

Table 10: Top level graph measures with selected amino acid descriptors

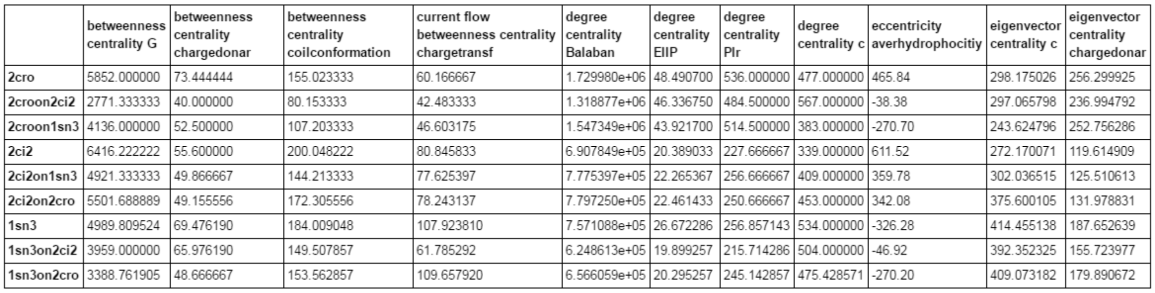


	betweenness centrality G	betweenness centrality chargedonar	betweenness centrality coilconformation	current flow betweenness centrality chargetransf	degree centrality Balaban	degree centrality EIIP	degree centrality Plr	degree centrality c	eccentricity averhydrophocitiy	eigenvector centrality c	eigenvector centrality chargedonar
2ci2	6416.222222	55.6	200.048222	80.845833	690784.90135	20.389033	227.666667	339.0	611.52	272.170071	119.614909

4 RESULTS

The process outlined in Section 3.1 is repeated for the proteins and their respective decoys. The proteins being studied are 2cro, 2ci2, and 1sn3. 2cro is partitioned into 5 cluster. 2ci2 is partitioned into 7 clusters. 1sn3 is partitioned into 9 clusters. Each of these proteins have 2 decoys that can be compared and the respective decoys are partitioned into the same number of clusters as the main protein resulting in Table 11.

Table 11: Top level graph measures with selected weighted descriptors



	betweenness centrality G	betweenness centrality chargedonar	betweenness centrality coilconformation	current flow betweenness centrality chargetransf	degree centrality Balaban	degree centrality EIIP	degree centrality PIr	degree centrality c	eccentricity averhydrophocitiy	eigenvector centrality c	eigenvector centrality chargedonar
2cro	5852.000000	73.444444	155.023333	60.166667	1.729980e+06	48.490700	536.000000	477.000000	465.84	298.175026	256.299925
2croon2ci2	2771.333333	40.000000	80.153333	42.483333	1.318877e+06	46.336750	484.500000	567.000000	-38.38	297.065798	236.994792
2croon1sn3	4136.000000	52.500000	107.203333	46.603175	1.547349e+06	43.921700	514.500000	383.000000	-270.70	243.624796	252.756286
2ci2	6416.222222	55.600000	200.048222	80.845833	6.907849e+05	20.389033	227.666667	339.000000	611.52	272.170071	119.614909
2ci2on1sn3	4921.333333	49.866667	144.213333	77.625397	7.775397e+05	22.265367	256.666667	409.000000	359.78	302.036515	125.510613
2ci2on2cro	5501.688889	49.155556	172.305556	78.243137	7.797250e+05	22.461433	250.666667	453.000000	342.08	375.600105	131.978831
1sn3	4989.809524	69.476190	184.009048	107.923810	7.571088e+05	26.672286	256.857143	534.000000	-326.28	414.455138	187.652639
1sn3on2ci2	3959.000000	65.976190	149.507857	61.785292	6.248613e+05	19.899257	215.714286	504.000000	-46.92	392.352325	155.723977
1sn3on2cro	3388.761905	48.666667	153.562857	109.657920	6.566059e+05	20.295257	245.142857	475.428571	-270.20	409.073182	179.890672

The data is normalized using an l_2 norm. Then we implement a hierarchical clustering method shown using a dendogram shown in Figure 8.

The protein 1sn3 is an outdated protein in the set. It is not as concerning that it is classified with the decoys. Protein 1sn3 and its respective decoys are removed from Table 11 as shown in Table 12.

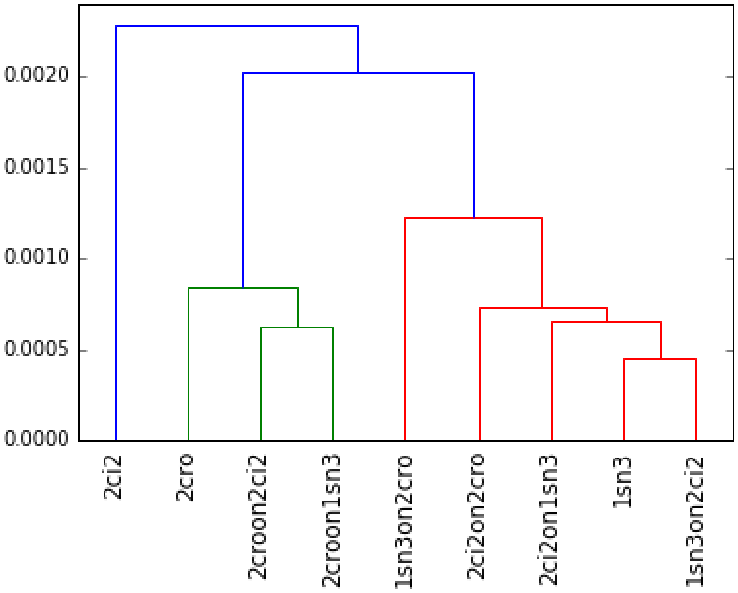
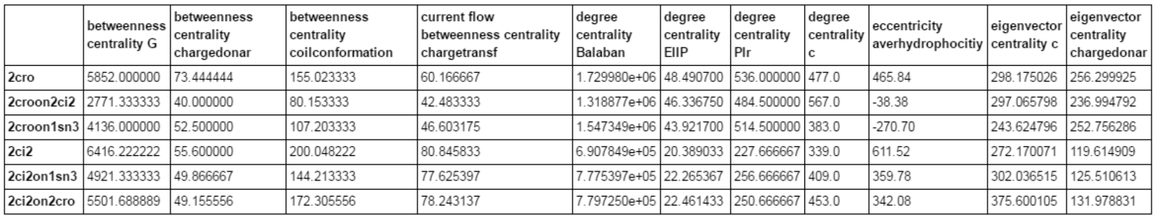


Figure 8: Dendrogram of the normalized l2 data for all proteins used

Table 12: Concatenated table of top level graph measures for 2ci2, 2cro, and respective decoys



	betweenness centrality G	betweenness centrality chargedonar	betweenness centrality coilconformation	current flow betweenness centrality chargetransf	degree centrality Balaban	degree centrality EIIP	degree centrality Plr	degree centrality c	eccentricity averhydrophocitiy	eigenvector centrality c	eigenvector centrality chargedonar
2cro	5852.000000	73.444444	155.023333	60.166667	1.729980e+06	48.490700	536.000000	477.0	465.84	298.175026	256.299925
2croon2ci2	2771.333333	40.000000	80.153333	42.483333	1.318877e+06	46.336750	484.500000	567.0	-38.38	297.065798	236.994792
2croon1sn3	4136.000000	52.500000	107.203333	46.603175	1.547349e+06	43.921700	514.500000	383.0	-270.70	243.624796	252.756286
2ci2	6416.222222	55.600000	200.048222	80.845833	6.907849e+05	20.389033	227.666667	339.0	611.52	272.170071	119.614909
2ci2on1sn3	4921.333333	49.866667	144.213333	77.625397	7.775397e+05	22.265367	256.666667	409.0	359.78	302.036515	125.510613
2ci2on2cro	5501.688889	49.155556	172.305556	78.243137	7.797250e+05	22.461433	250.666667	453.0	342.08	375.600105	131.978831

The data is normalized using an l_2 norm. Then we implement a hierarchical clustering method shown using a dendogram shown in Figure 9.

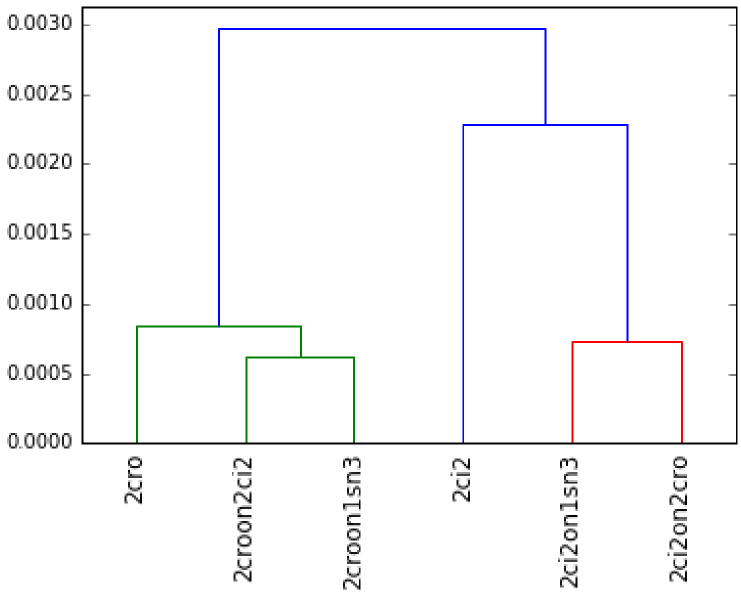


Figure 9: Dendrogram of the normalized l2 data for 2ci2, 2cro, and respective decoys

The quantification of proteins with a nested graph model that was partitioned by spectral clustering shows that there are distinct characteristics that belong to the sets of proteins. It is shown that the selected proteins are grouped with their decoys, but are distinctly different than the respective decoy.

5 CONCLUSION

From the first result in Figure 8, we see that the proteins are grouped together with their respective decoys. With the knowledge that the protein 1sn3 is obsolete, it is no surprise that it is grouped with the decoys. The removal of 1sn3 and its respective decoy proteins yields the second result, Figure 9. It is shown that the remaining proteins, 2cro and 2ci2, are completely separated from their respective decoys. This is shown in both cases.

This model shows that incorporating a different sequence partitioning method for proteins and their decoys in a graph theoretic model yields results that groups well but at the same time, keeps the proteins separated from their respective decoy.

Further advancement of this method could be used by researchers to refine protein structure prediction algorithms. The nested graph model with vertex weights derived from features of protein structures shows promise as an added tool for the advancement of protein science.

BIBLIOGRAPHY

[1] *AAindex*, http://www.genome.jp/aaindex/

[2] Balaban, A. T., *Distance Connectivity Index*, Chem. Phys. Lett. 89, 399-404, 1982.

[3] Beineke, Lowell W. and Wilson, Robin J., *Topics in Algebraic Graph Theory*, 1st edition, Published by Cambridge University Press. 2004.

[4] *BIOPYTHON*, http://biopython.org/wiki/Documentation

[5] Bondy, J.A., and Murty, U.S.R., *GTM Graph Theory*, Published by Springer, 2008.

[6] Brandes, Ulrik, *On Variants of Shortest-Path Betweenness Centrality and their Generic Computation*, Department of Computer & Information Science, University of Konstanz, 12 Nov. 2007.

[7] Brandes, Ulrik and Fleischer, Daniel, *Centrality Measures Based on Current Flow*, STACS 2005, LNCS 3404, pp. 533544, 2005.

[8] Cosic, I., *Macromolecular bioactivity: is it resonant interaction between macromolecules? Theory and applications*, IEEE Trans Biomed Eng. 41(21):1101-14, Dec 1994.

[9] Charton M, Charton BI, J Theor, *The dependence of the Chou-Fasman parameters on amino acid side chain structure*, Biol., 102(1):121-34, May 7 1983.

[10] Cid H, Bunster M, Canales M, Gazitua F, *Hydrophobicity and structural classes in proteins*, Protein Eng., 5(5):373-5, July 1992.

[11] Cooper, Geoffrey M. and Hausman, Robert E., *The Cell A Molecular Approach*, Published by ASM Press, 2007.

[12] Edwards, Stajich, and Hansen, *Bioinformatics Tools and Applications*, Published by Springer, 2009.

[13] Estrada, Ernesto, *The Structure of Complex Networks Theory and Applications*, 1st edition, Published by Oxford University Press, 2011.

[14] Estrada, Ernesto and Rodriguez-Velazquez, Juan A., *Subgraph Centrality in Complex Networks*, Physical Review E 71, 056103, 2005.

[15] Everitt, B. S. and Skrondal, A., *The Cambridge Dictionary of Statistics*, Published by Cambridge University Press, 2010.

[16] Freeman, Linton C., *Centrality in Social Networks Conceptual Clarification*, Social Networks, 1, 215-239, 1978/79.

[17] Gross, Jonathan and Yellen, Jay, *Graph Theory and Its Applications*, 1st edition, Published by CRC Press LLC., 1998.

[18] Hogeweg, Paulien, *The Roots of Bioinformatics in Theoretical Biology*, PLoS Comput Biol 7(3), 2011.

[19] Horn, Roger A. and Johnson, Charles R., *Matrix Analysis*, Published by Cambridge University Press, 2013.

[20] Knisley, Debra, Knisley, Jeff, and Herron, Andrew Cade, *Graph-Theoretic Models of Mutations in the Nucleotide Binding Domain 1 of the Cystic Fibrosis Transmembrane Conductance Regulator*, Published by Hindawi Publishing Corporation, 12 March 2013.

[21] Knisley, Debra J. and Knisley, Jeff R., *Predicting proteinprotein interactions using graph invariants and a neural network*, Computational Biology and Chemistry 35, 108113, 2011.

[22] Knisley, Debra J. and Knisley, Jeff R., *Seeing the results of a mutation with a vertex weighted hierarchical graph*, BMC Proc., 8(Suppl 2): S7, 2014.

[23] Luxburg, Ulrike von, *A Tutorial on Spectral Clustering*, Statistics and Computing, 17 (4), 2007.

[24] Martin, Charles H., *Spectral Clustering: A Quick Overview*, 2012. `https://calculatedcontent.com/2012/10/09/spectral-clustering/`

[25] *NetworkX*, `http://networkx.readthedocs.io/en/stable/reference/algorithms.html`

[26] Paccanaro, Alberto, Casbon, Mansoor, James A. , Saqi, A. S., *Spectral clustering of protein sequences*, Nucleic Acids Research, 34 (5): 1571-1580, 2006.

[27] *Pandas*, `http://pandas.pydata.org/`

[28] *The Protein Data Bank*, `http://www.pdb.org`

[29] Samudrala, Ram and Levitt, Michael, *Decoys R Us: A database of incorrect conformations to improve protein structure prediction*, Protein Science, 9:1399-1401, 2000.

[30] Samudrala, Ram, *Decoys 'R' Us*, Computational Biology Research Group, `http://ram.org/compbio/dd/`

[31] *scikit-learn*, `http://scikit-learn.org/stable/`

[32] *SciPy*, `https://www.scipy.org/`

[33] *'sklearn.cluster'.SpectralClustering*, 2010 - 2016, scikit-learn developers (BSD License).

[34] Trinajsti, Nenad, *Chemical Graph Theory Volume Ić*, Published by CRC Press, 1983.

[35] Vishveshwara, Saraswathi, Brinda, K. V. and Kannany, N., *Protein Structure: Insights From Graph Theory*, Journal of Theoretical and Computational Chemistry, Vol. 1, No. 1, Published by World Scientic Publishing Company, 2002.

[36] Yan, Yan, Zhang, Shenggui, and Wu, Fang-Xiang, *Applications of graph theory in protein structure identification*, Proteome Sci., 9(suppl 1): S17, 2011.

[37] Zhao, B., Carson, M., Ealick, S.E., Bugg, C.E, *Structure of Scorpion Toxin Variant-3 at 1.2 Angstroms Resolution*, 31 Jan. 1994. `http://www.rcsb.org/pdb/explore/obsolete.do?obsoleteId=1SN3`,

VITA

WALTER G. WITT

Education:

M.S. Mathematical Sciences, East Tennessee State University

Johnson City, Tennessee 2017

Graduate Certificate Math Modeling Biosciences, East Tennessee State University

Johnson City, Tennessee 2017

M.Ed Secondary Education, Emory & Henry College,

Emory, Virginia 2009

B.S. Mathematics, Emory & Henry College,

Emory, Virginia 2008

Professional Experience:

Graduate Assistant, East Tennessee State University,

Johnson City, Tennessee, 2015–2017

High School Teacher

North Carolina & Virginia, 2009–2015

Presentations:

Quantifying Misfolded Protein Structures Using Graph Theory

Appalachain Student Reseach Forum, Johnson City, TN, 2017

Quantifying Misfolded Protein Structures Using Graph Theory

MCBIOS, Little Rock, AR, 2017