Graph Construction for Modeling Protein 3D Structures - Consequences and Improvements

Amy Wagaman [*]

Abstract
Researchers often model folded protein 3D structures as graphs with amino acids as the vertices and edges representing contacts between amino acids. Many possible constructions exist based on whether the graph is made using all atoms or only C-alpha atoms or only C-beta atoms, deciding what counts as contacting for determining the graph edges, and deciding to ignore or not ignore trivial contacts from amino acids close in the protein sequence. However, there is no consensus about what construction to use or what the major issues are with each construction in the literature. We investigate different constructions and examine their effect on various graph measures. We also consider the "small-world" network model for proteins, discuss its validity under the different constructions, and discuss random protein graph generation.

Key Words: networks, proteins, small-world, graph theory, random graph generation

1. Introduction

Understanding the mechanisms in protein folding and predicting the three-dimensional structure of a protein are challenging problems. Research groups use physical models, simulations, and templates (portions of known proteins similar to the one under investigation) in procedures to get the best realistic protein structure prediction that they can. Other groups use known protein three-dimensional structures to try to shed light on the protein folding problem. In many cases, the researchers model the folded protein three-dimensional structures as a graph with amino acids as the vertices and edges representing contacts between nearby amino acids. Many possible graph constructions exist due to different representations of the protein, and there is no consensus about which construction to use in the literature. After some background information on proteins and graphs, we consider various graph representations from the literature with their applications.

A graph is a collection of vertices and edges (V, E), where an edge is a 2-element subset of the set V indicating a connection between those vertices. An edge may be directed or undirected and may or may not have a weight. The graphs we consider are undirected graphs. Most of these graphs are simple, i.e. they do not have multi-edges (meaning there is either no edge or only one edge between any two vertices), but one representation allows multi-edges. Our constructions do not have self-edges. Each amino acid in the protein sequence is a vertex and edges reflect that the amino acids are in contact in their three-dimensional folded structure. To determine contacts, a distance between amino acids is calculated, and if it is below a threshold, the amino acids are in contact. The most common distance used (and what we use) is Euclidean distance between atoms while folded.

Proteins are chains of amino acids that fold into a specific shape to perform a job. Each amino acid has a backbone, the same for all amino acids, except glycine, and a sidechain which differs between amino acids. In the backbone, there are two carbon

[*] Department of Mathematics, Amherst College, P.O. Box 5000, Amherst, MA 01002

atoms - commonly called Carbon-alpha (C-alpha) and Carbon-beta (C-beta). These atoms are convenient points of reference. When dealing with distances on this atomic scale, the distance unit is the Angstrom (A). One additional consideration is that because the amino acids are ordered, you expect connections between amino acids near in sequence. In some applications, these contacts are considered trivial, and may be removed. For example, you may not consider a contact to be a true contact unless the amino acids are more than two apart in sequence. Finally, protein structures in their folded state are determined using chemical techniques and the results are entered into a freely available database (RCSB) (1). For our work, we use the PDB (protein databank) files associated with our proteins downloaded from RCSB.

We offer a representative literature review of applications where graph representations of proteins with amino acids as the vertices have been used. For each case, we note the application and the graph construction used (or implied) including what atoms were used as references to determine distances, the distance cutoffs used, and whether or not a filter was used to remove trivial contacts. For notation, a C-Alpha protein representation refers to only C-Alpha atoms being used to determine distances. Other representations to determine distances are C-Beta and all-atom representations. The literature review is summarized in Table 1, with general patterns discussed in the text.

Table 1: Example Protein Graphs In the Literature.

First Author	Year	Rep.	Dist.	Filter	Application	Citation
Rodionov	1994	C-Beta	-	No	contact substitution	(12)
Plaxco	1998	All-atom	6A	No	folding rate prediction	(11)
Gromiha	2001	C-Alpha	8A	Yes	folding rate prediction	(4)
Vendruscolo	2002	C-Alpha	8.5A	No	small-world graphs	(13)
Ivankov	2003	All-atom	6A	No	folding rate prediction	(6)
Greene	2003	All-atom	5A	Yes	protein graphs	(3)
Jung	2005	C-Alpha	8A	No	unfolding rate prediction	(7)
Krishnan	2008	C-Alpha	6A	Yes	protein graphs	(8)
Habibi	2010	C-Alpha	8A	No	protein graphs	(5)

Protein graphs (contact maps) have been used since the 1970s (12). As seen in Table 1, they continue to be used in current research. The most common representation is C-Alpha. Distances typically range from 5-10A. Filters are not universally used. In our examples, the filters occurred at different sequence separations. For example, amino acids needed to be more than 2 amino acids apart for contacts to count in Krishnan's work (8), but Gromiha focused on long-range contacts more than 12 amino acids apart (4). All-atom graphs may allow multi-edges. Finally, we see from the applications that much of the related work deals with protein folding, but there has been a recent shift towards work with proteins as graphs.

As work has turned to understanding the protein graphs as graphs, little attention has been paid to how the various constructions affect values typically calculated for graphs. In this work, we consider the effects of these various constructions on graph properties and implications for generating random graphs that behave like protein graphs. Our outline of the paper follows. First, we introduce relevant graph definitions in Section 2. We introduce our protein dataset in Section 3. Our methods for protein graph construction are presented in Section 4. Next, we show the impact of the various construction methods on the "small-world" property of protein graphs in Section 5. We then discuss the impact of the different constructions on some graph concepts in Section 6. In Section 7, we highlight some preliminary results of work to develop a random protein graph generator, as well as show that current random graph generators (even small-world generators) do not yield realistic protein-like graphs. Finally, we conclude with discussion, some suggestions regarding the constructions, and future work in Section 8.

2. Definitions of Graph Concepts

As seen in the Introduction, some researchers computed graph concepts for protein graphs and evaluated their use in understanding protein folding. We define the properties we examine in this section. Note that many graph concepts do not have appropriate adjusted computations for graphs with multi-edges. As a result, our primary focus is to compare the simple graph constructions. The vertices of our graphs are the amino acids, labeled from 1 to n in sequence order, and there are a total of m edges determined by contacts, where m and n depend on the protein. A typical representation of the graph is it's adjacency matrix, A. The matrix A is n by n and the ij^{th} entry in the matrix is the number of edges between vertex i and vertex j. For further details or as an additional reference on graph basics, the reader is directed to (9). We begin our definitions with the degree of a vertex.

2.1 Degree, Number of Edges/Contacts, and Degree Distribution

For each vertex, q_i, $i = 1, \ldots, n$, is the degree of the vertex. This is simply the number of edges which connect to the vertex, and is easily computed as $q_i = \sum_{j=1}^{n} A_{ij}$. For the protein graphs, this is equal to the number of contacts determined for each amino acid in the protein sequence.

One of the most important characteristics of a graph is its degree distribution. We let p_q be the fraction of vertices in the graph with degree q. For simple graphs, the upper limit on q is $n-1$, and so $\sum_{q=0}^{n-1} p_q = 1$. It is not uncommon for the degree distribution to follow a power law, so that $p_q = Cq^{-\alpha}$, $2 < \alpha < 3$, and where C is an appropriate constant. Finally, degree is also a measure of centrality. A vertex is more central if it has more connections. Not all neighbors are equivalent however, so it is a good idea to consider alternative centrality measures (9).

2.2 Average Path Length (APL)

Shortest path length is the minimum number of edges that must be traversed to go between two vertices. Average path length, APL, is the average of all the shortest path lengths when considering all pairs of vertices.

2.3 Clustering Coefficient (CC)

There are several non-equivalent definitions of the clustering coefficient. In general, the clustering coefficient is a measure of how tightly clustered the graph is. For the first definition, it is computed as 6 times the number of triangles divided by the number of paths of length 2 in the graph. In other words, the clustering coefficient is the number of triangles out of the number of possible triangles starting from two connected sides. We decided to look at a second definition of the clustering coefficient because we want to use graph concepts to identify important individual vertices. This measure is computed for each vertex, and then averaged across all vertices. For each vertex, v, the local clustering coefficient is the number of pairs of neighbors of v which are also connected divided by the number of pairs of neighbors of v (i.e. it is analogous to the first definition, just localized to each vertex), with vertices ignored which have fewer than 2 neighbors. Then, the average is taken over all vertices that have at least 2 neighbors. We refer to the clustering coefficient as the average clustering coefficient, or CC. As we stated, these two definitions are not equivalent, and we use the latter.

2.4 Small-world Properties

Many social and biological networks have been found to be much more highly clustered than random graphs with similar numbers of vertices. Graphs with high clustering coefficients and slightly higher values for average path length (but still low overall) as compared to random graphs of the same size are often termed “small-world” graphs (15). Researchers have shown protein graphs exhibit small-world tendencies (3) (13).

2.5 New Graph Property: Contact Distribution

Contact distribution is a new graph property we developed for protein graphs due to the natural ordering of vertices. When we considered degree distribution, we wanted to know how many vertices had each degree value. Contact distribution is the distribution of weights along the edges in the graph, where the weights are the sequence separation between amino acids that are in contact. In other words, if the edges were given weights according to how far apart their vertices were in sequence, we want to understand how many edges have each possible weight value, and study the weight distribution across the graph. Let r_i be the fraction of contacts/edges that occur at a sequence separation of i (out of the $n - i$ possible edges at that sequence separation distance). Each r_i is between 0 and 1, and a simple rescaling $s_i = r_i * (n - i)/m$ allows for a constraint that $\sum_{i=1}^{n-1} s_i = 1$. In this rescaling, s_i is the fraction of existing edges in the graph that occur at sequence separation i.

2.6 Other Graph Measures

Many other graph measures exist, and results for these measures are omitted in this paper, but will be presented in future work. Briefly, we highlight some measures we considered. We examined graph stability through edge removal impact probability (ERIP). This measure was originally proposed in (7), and is computed based on the average path length but with varying percentages of edges removed from the graph. In our work, we modify this procedure and study similar relationships to those in (7). We also examined several measures of centrality including eigenvector centrality, betweenness centrality, and closeness centrality (9). Our centrality results are promising because they may indicate important connections that occur in the protein folding process. Finally, we also studied some connectivity measures, which may be related to packing/size protein properties.

3. Protein Data

Our data consists of 127 cases - distinct proteins which were collected to create a database of proteins with thermodynamic and kinetic information available. The database is currently maintained by Amherst College. Preliminary database details are available in (14). For the analysis in this paper, the PDBs of the proteins were downloaded from RCSB (1) and processed using a Perl script to obtain the protein graphs under the methods described in the next section. The graph concepts were then computed from the protein graphs using R, the igraph package, and original code, and compared to other variables in the database. Reproducing the graph concept analysis on a larger set of proteins sampled from RCSB is an area for further investigation, but not all proteins have experimental thermodynamic and kinetic data available.

To get a sense of the data, we consider a few descriptive statistics. For the 127 proteins, the average size is 107.5 amino acids, while the median size is 86 amino acids. Twenty-eight of the proteins are multi-state folders, and 65 are two-state folders. We have folding rate constants for 115 of the proteins and unfolding rate constants for 49 proteins. The average helical content of a protein in the data set is 22.48 percent (median 16 percent) and average beta sheet content is 23.87 percent (median 26 percent). Finally we have all four structural classes represented: 28 are class α, 36 are class $\alpha + \beta$, 8 are class $\alpha\beta$, 48 are class β and 7 have unknown class (or are fragments).

4. Methods for Protein Graph Construction

Recall that in the graphs we consider, each amino acid in the protein is a vertex, labeled according to the amino acid sequence. Edges are added between amino acids which are in contact when the protein is in its folded three-dimensional native structure. There are several aspects of the graph construction that we examine: the protein representation (atoms used to determine distance), the distance cutoff, and filters, which are used to eliminate trivial (and other) contacts.

4.1 Protein Representation

We consider three different protein representations to determine distances between amino acids in the three-dimensional structure of the protein. The first is the common C-Alpha to C-Alpha representation, where only C-Alpha atoms are used. We refer to this method as CA (C-Alpha).

The second is an all-atom representation where all atoms except hydrogens are considered. For any two amino acids, all pairs of non-hydrogen atoms are examined and the minimum distance between the pairs is set as the distance between the amino acids. This method is referred to as AA (all-atom). Hydrogens are not considered because their positions are often unresolved or are unclear in the three dimensional native structure determined by X-ray crystallography or NMR.

Finally, we use the same all-atom representation, but count the number of pairs of non-hydrogen atoms whose distance is less than our cutoff distance for each pair of amino acids. In this final method, multi-edges may result between amino acids, so we refer to the method as MC (multiple contacts).

4.2 Distance

We examined distance cutoffs from 6-12 A in steps of .5. We did in-depth examinations of graph concepts at 6, 8, and 10 A, though most of the patterns we found are similar for each distance and we focus on 8 A for the discussion.

4.3 Filter

Filters are designed to remove trivial (and other) contacts from the graph. We already do not allow self-edges, so the diagonal of the adjacency matrix for each graph is set to 0. Filters remove subsequent diagonals in the adjacency matrix, moving out from the main diagonal. We set up our filter to be indexed by a parameter k. $k = 1$ means the first diagonal is removed (the main diagonal) so this is equivalent to the original adjacency matrix. $k = 2$ means that the second diagonal is removed (all values set to 0)(so the main diagonal, and first diagonal on either side), etc. Thinking about this in the protein context, a filter at value k means that amino acids must be at least k apart in sequence in order for the contact to count. Suppose we have amino acid i, $i < n - 2$, and a filter of $k = 2$. Then, this means that amino acids i and $i + 2$ could be in contact, but i and $i + 1$ would not be allowed to be.

We note an important cutoff for choice of k. Alpha helices (an important part of secondary structure in proteins) have natural contacts at amino acids i and $i + 4$ all along the helix. So at a filter value of $k = 5$ or higher, those natural contacts have been removed. We examined filters from 1-20 across our different distances and methods, though at times we focus on filters of $k = 1, 4, 10$. Those filters were chosen to compare the original graph ($k = 1$), a graph with trivial contacts removed ($k = 4$) but where alpha helix contacts were retained, and a graph where only long-range contacts remained ($k = 10$).

5. When Are Protein Graphs Small-World?

Several researchers have identified protein graphs as small-world graphs (3)(13). We examine the effect of the various construction methods, distance cutoffs and filters on the graphs in a small-world context. We found that the choice of distance cutoff did not influence the overall pattern, so we fix it here at 8A. Figure 1 shows the average clustering coefficients plotted against the average path length for our data under the AA and CA methods and at filters of $k = 1$, $k = 4$, and $k = 10$. Clearly, the non-filtered graphs ($k = 1$) are the ones that are small-world. Also, the AA construction appears to have lower average path lengths and slightly higher clustering coefficients than the CA method, which is expected. In summary, to pick a construction that is small-world, one must use the non-filtered graph from an AA representation at a reasonable distance. We note that our clustering coefficients are higher than those in (3), but their graphs were constructed with a distance cutoff of 5A, compared to ours at 8A.

It is clear from Figure 1 that the application of filters destroys the small-world property. This is intuitive because the natural ordering of amino acids and methods of graph construction result in sequence neighbors being graph neighbors, and the clustering coefficient should plummet when those trivial contacts are removed. However, we also note that the CA construction seems to be much more sensitive to the filter than the AA method. One can also note that for the CA method and $k = 10$ filter, a group of proteins has zero as the value of the clustering coefficient, so

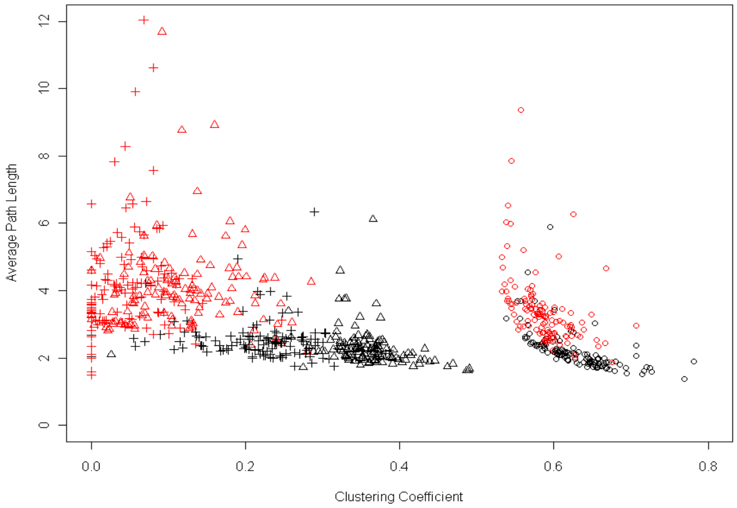

Figure 1: Scatterplot of average clustering coefficient vs. average path length for the AA (black) and CA (red) methods at 3 different filter levels at 8A. Circles are k=1 filter (i.e. original graph), triangles are k=4, and plus signs are k=10.

that effectively all connected triples have been dismantled, or so many connections have been removed that no amino acid has at least 2 neighbors.

6. Effects on Graph Concepts

6.1 Degree Changes

For our 127 proteins, we show how degree changes between the methods, over the distances, and as we apply filters. The results are intuitive. We compute the average and standard deviations of the mean degrees for our proteins at each distance, filter, and method (for CA and AA) and report them in Table 2 for comparison.

Table 2: Averages (SDs) of the mean degrees for 127 proteins under different representations, distances, and filters.

Rep.	AA			CA		
Filter/Dist.	6A	8A	10A	6A	8A	10A
$k = 1$	12.54 (2.56)	21.39 (3.80)	30.38 (5.71)	5.48 (1.16)	9.66 (1.67)	16.50 (2.95)
$k = 4$	7.64 (2.41)	15.61 (3.78)	24.54 (5.68)	1.92 (0.96)	4.86 (1.64)	10.95 (2.91)
$k = 10$	5.30 (2.19)	11.39 (3.82)	18.58 (5.80)	1.33 (0.78)	3.30 (1.53)	7.74 (2.95)

It is clear that at each distance and for each filter, the degrees for CA graphs will be less than or equal to those for AA graphs. Additionally, degrees can only decrease or stay the same as we apply filters. For a distance increase however, degrees are likely to increase. The general pattern to these differences in the graphs is not surprising. But what other changes are there? Should we compute other graph measures without accounting for filters? Can we say anything about the relationship between edges in the different graphs? What do these relationships tell us about protein folding? These are example questions we seek to address.

6.2 Relationship between Absolute Contact Order and Folding Rate Constants

Contact order is correlated with the natural log of the folding rate constant of proteins. An MC graph is used for the computation of contact order (11). We demonstrate that the AA graph results in very similar relationships to the natural log of folding rates, and we know computation of the AA graph is faster than the MC graph. Here, we examine how absolute contact order computed on MC, AA, and CA graphs at each distance and filter relates to the natural log of folding rate for the proteins in our data set, after describing the computation of contact order. Contact order was proposed in 1998, and is easily constructed from a protein graph. In the original notation, L is the chain length of the protein, N is the total number of contacts and $\Delta S_{i,j}$ is the sequence separation between residues i and j for contacting residues. Then, using an MC graph representation with $k = 1$ filter and distance cutoff of 6 A, contact order is computed as:

$$CO = \frac{1}{LN} \sum^{N} \Delta S_{i,j}, \tag{1}$$

where the sum is over all contacting residues found based on the distance computation and cutoff chosen (11). Contact order was modified by multiplying by L to form absolute contact order (ACO), and ACO was found to perform better than contact order in predicting folding rate constants (6). For our work, we compute ACO. For this part of the analysis, we use a subset of 50 proteins where we had the folding rate available, as reported in (6) or (11). Table 3 contains correlations between the natural log of the folding rate constant and the ACO under each representation, at the three different distances for filters from $k = 1$ (no filter) to $k = 10$. For reference, in the ACO paper, the observed correlations between ACO and lnkf were: -.51 for two-state folders only, -.78 for multi-state folders only, and -.74 over all proteins considered (6). Our data set has a mixture of two-state and multi-state folders. We have results for filters 1-20, but because the pattern is clear, we omit the results for filters 11-20.

Considering each distance in turn, a few things are clear. First, the AA correlations with the natural log of the folding rate are very similar to the MC correlations, while CA appears to have different (weaker) correlations (though at 10 A, the difference is minor). Also, as the amount of filtering increases (k increases), the correlations tend to decrease. Most distance/representation combinations have a correlation drop around $k = 5$. Recall that starting at $k = 5$, the i to $i + 4$ contacts expected in alpha-helices are removed. We note that the correlations are also fairly stable over the different distances within each method, except CA. Overall, it appears a filter to remove trivial contacts, say $k = 3$ or $k = 4$, does not significantly impact the resulting ACO correlation with the natural log of the folding rate, and that distance chosen really only affects results with a CA graph representation. Additionally, based on our work, we have observed a relationship between the number of contacts in the graphs under the different methods, especially once trivial contacts ($k = 4$ or less) have been removed. We believe these relationships shed some light on protein folding and packing, and have publication work in progress.

Table 3: Absolute Contact Order correlation with LNKF (natural log of the folding rate constant) across filters $k = 1$ to $k = 10$ (2 significant digits) at three distances (6, 8, and 10 A) for the three different graph representations.

Distance/Rep.	6 A			8 A			10 A		
Filter	MC	AA	CA	MC	AA	CA	MC	AA	CA
$k = 1$	-.67	-.67	-.58	-.69	-.67	-.62	-.69	-.66	-.65
$k = 2$	-.67	-.67	-.57	-.69	-.67	-.62	-.68	-.66	-.65
$k = 3$	-.68	-.67	-.60	-.68	-.67	-.61	-.68	-.66	-.65
$k = 4$	-.67	-.67	-.59	-.66	-.67	-.61	-.67	-.66	-.65
$k = 5$	-.61	-.64	-.53	-.62	-.66	-.56	-.65	-.66	-.64
$k = 6$	-.58	-.59	-.49	-.59	-.65	-.54	-.63	-.65	-.61
$k = 7$	-.58	-.58	-.49	-.59	-.63	-.53	-.61	-.64	-.59
$k = 8$	-.59	-.59	-.49	-.58	-.61	-.53	-.60	-.63	-.58
$k = 9$	-.58	-.58	-.48	-.58	-.61	-.52	-.59	-.62	-.57
$k = 10$	-.58	-.58	-.47	-.58	-.60	-.52	-.59	-.61	-.57

6.3 Average Path Length

We considered average path length in our small-world discussion. Our analysis shows that as expected, CA graphs have longer average path lengths than AA graphs at the same filter and distances. Longer distances mean shorter average path lengths, and higher filters mean longer average path lengths. Also, generally, the AA graph with a $k = 10$ filter has a shorter average path length than the CA graph with no filter, $k = 1$. At 10 A, most of the path lengths for AA or CA graphs are between 2 and 4. Even when we look at 6A for CA graphs, the longest path lengths are between 10 and 15, and most are between 4 and 7. Considering the size of some of these graphs, that is impressive, and results for AA graphs are consistent with the small-world belief we examined earlier.

Briefly, we consider the relationship between average path length and graph size focusing on differences between AA and CA methods at 8A. Figure 2 is a scatterplot showing the relationship with no filter applied. Average path length does increase slightly as graph size increases, as expected. For the AA method, applying a filter does not increase the average path length much. At 8 A, the average increase at $k = 4$ is only .11 and at $k = 10$ this goes up slightly to .34, but is still less than one additional edge, and is similar for the other distances we examined. CA graphs have larger increases in average path length as filters are applied. In CA graphs, at 8 A, for $k = 4$, average path lengths are on average, .64 longer than their $k = 1$ counterparts, and $k = 10$ average path lengths are .85 longer on average. This is still less than a one edge increase, on average. So while filters do increase the path length, the biggest differences are due to the method. The average difference between AA and CA average path lengths at 10 A is .63, at 8 A is 1.22, and at 6 A is 2.57. Thus, at higher distances, the method difference is not as pronounced.

6.4 Clustering Coefficient

We also considered the clustering coefficient in our small-world discussion. We already know that applying a filter reduces this measure significantly. Our results show that notably, even at 10 A, some of the proteins have a 0 for their CC under

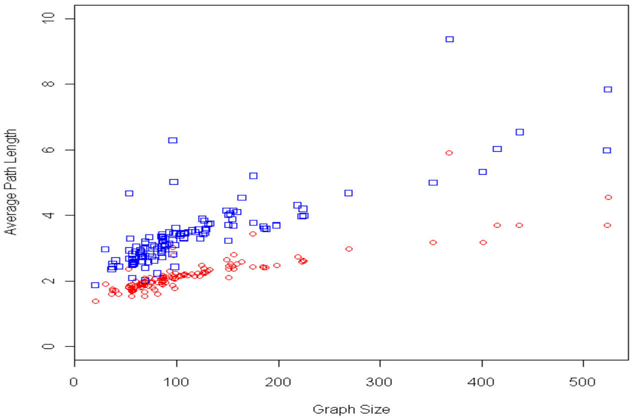

Figure 2: Graph size versus average path length of each protein in the data set at 8 A with no filter $k = 1$. CA average path lengths are blue squares, while AA average path lengths are red circles.

a CA construction at $k = 10$ (this occurs even more at 6A). Clustering coefficients increase as distance increases, and the methods do not result in terribly different clustering coefficients if no filter is present ($k = 1$). Once filters are applied, AA graphs have higher clustering coefficients than CA graphs, and this is more pronounced at lower distances. Relating back to the small-world discussion, we already stated that the breakdown of the small-world property when filters are applied is due to the drops in clustering coefficients, not changes in average path lengths. This has some implications for using the clustering coefficient as a measure of how tightly clustered protein graphs are. For long-range filters, and especially if using the CA method, the clustering cofficients drop to near 0 and there is not much variability in their values. Hence, it might be best to only consider the clustering coefficient without filters applied, or to develop a new way to quantify long-range triangle neighbor relationships.

6.5 Summary of Results on Other Measures

We also examined the other graph measures we introduced in Section 2. Complete results will be presented in forthcoming work, but we highlight a few aspects related to centrality here. Centrality scores can increase or decrease as filters are applied. Examining change or lack thereof in centrality across filters may allow important vertices (amino acids) to be identified. These scores are also very different for CA graphs at short distances than AA graphs and vertices with a high score in the sparse CA graph might indicate a folding contact that must be achieved early in the folding process.

7. Steps Toward Constructing a Random Protein Graph Generator

In this section, we give brief background on random graph generation, especially small-world graph generation, and investigate if these graphs can mimic protein graphs. The main feature we try to mimic is the protein contact distribution. Developing a model to create protein graphs may shed light on protein folding depending on what properties must be enforced in order to achieve realistic protein graphs.

7.1 Existing Methods for Random Graph Generation

The most basic random graph models are $G(n, m)$ and $G(n, p)$ models. The $G(n, p)$ model is often referred to as the Poisson model, because in the limit of large n, the degree distribution that results is Poisson (9). Obviously, being restricted to a Poisson degree distribution is a limitation, and other random graphs have been developed that can model any degree sequence, such as the configuration model (9). All of these models however, do not have high clustering coefficients, which often occur in real-world graphs. Alternative models, including small-world models, were developed to achieve that property.

Generating a small-world graph can be done in several ways. In the original proposal of Watts/Strogatz, small-world graphs are generated by starting with a ring of vertices. The vertices are all connected to some number of neighbors f, and each edge has the same fixed chance of being re-wired (probability w). Using this generation mechanism, long-range connections can be introduced, which decreases the average path length. However, the clustering coefficient remains strong due to the starting neighbor connections (15). A variant is to keep all the original connections and add a few long-range ones with probability w.

Other models for generating small-world graphs exist. Nguyen and Martel describe Kleinberg's model as well as a generalization (10). In Kleinberg's model, a grid is the basic starting unit for the graph. Each vertex is connected to its neighbors on the grid. Then, q long-range connections are added based on a probability that is inverse squarely proportional to the grid distance between each pair of vertices. Generalizations are made to models that start with a grid and add q long-range edges under other probability distributions (that can be vertex specific) (10). Many other random graph models exist, including models for directed graphs, growing graphs, etc. For a broad review of graph generators, see (2).

7.2 Protein Contact Distributions

The main feature that makes protein graphs interesting is the natural order of the vertices, and its consequences. If sequence separation is set as the edge weight for a protein graph, then we define the distribution of the edge weights as the contact distribution. The contact distribution may be scaled in one of two ways - either consider the number of edges at each sequence separation as a fraction of the maximum possible at each sequence separation $(n - 1)$, or as a fraction of existing edges. Contact distributions have interesting shapes due to protein folding patterns. An example contact distribution under the AA construction at 8A for PDB 1APS with no filter is shown in Figure 3. The graph has 98 vertices and 1658 edges. The rescaling was chosen as a fraction of existing edges. Unlike a degree distribution, which can have significant shape changes (not just rescaling) due to filters, the effect of a filter k on a contact distribution is just to remove the first $k - 1$ sequence separations from consideration, and rescale the distribution if based on fraction of existing edges.

The contact distribution example from 1APS shows interesting "humps". These humps occur due to the formation of long-range contacts. For example, the first hump in Figure 3 occurs around sequence separation 30. This might be because amino acid 12 was in contact with amino acid 42, which suggests amino acid 11

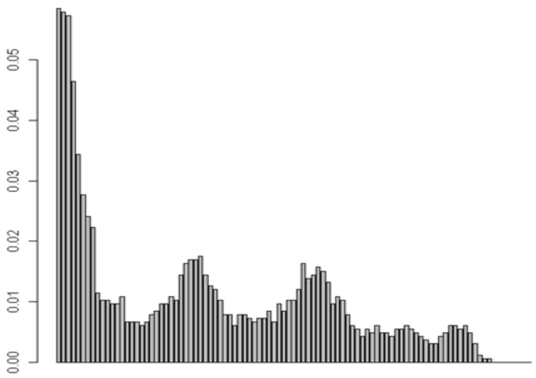

Figure 3: Contact distribution of PDB 1APS under AA construction at 8A with no filter ($k = 1$).

might be in contact with amino acids 42 or 43, and that amino acid 12 might be in contact with amino acid 43, etc. There might also be multiple neighborhoods involved. For example, it might be a contact between amino acids 12 and 42 and another contact between amino acids 25 and 55 and related connections that cause the hump.

It is not difficult to compute contact distributions for graphs generated from random graph generators. An example contact distribution from a Watts/Strogatz ring model (igraph function watts.strogatz.game(1, 100, 16, .3)) with 100 vertices, 1600 edges, and a rewiring probability of .3 is shown in Figure 4. The number of vertices and edges were chosen to be similar to the graphs from 1APS. The rewiring probability was chosen to provide a degree distribution similar to that of 1APS. Even with these similar settings, the contact distribution from the random small-world graph does not look at all like the contact distribution of the protein graph.

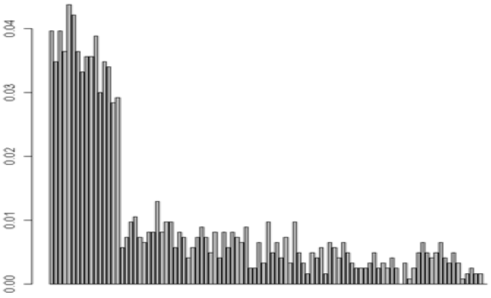

Figure 4: Contact distribution of a small-world graph generated from a Watts/Strogatz ring model (100 vertices, 1600 edges, rewire probability =.3).

Currently, we are investigating ways of quantifying the differences - defining a hump, number of humps, length of humps, etc. It is clear however that current random graph generators do not provide contact distributions that mimic protein graphs, even though their number of edges, vertices, degrees, clustering coefficients, and average path lengths can be similar. This leads to some natural questions. How can we obtain random protein graphs? Can we put protein graphs in a framework where they are a subset of small-world graphs (bur with ordered vertices)? What do we learn about protein folding/packing from looking at how we make random protein graphs?

7.3 Considerations for Random Protein Graphs

7.3.1 Using a Grid/Ring Building Block

The ring/grid building block of the small-world models considered as examples is a good starting point. As seen in the example protein contact distribution, Figure 3, there are a number of connections at small sequence separations. However, the drop-off is pretty extreme, at around sequence separation 7-10 in most protein graphs we examined. The grid/ring basis needs to accurately capture the drop-off. This has several implications if starting from a Watts/Strogatz or Kleinberg model. The Watts/Strogatz model needed is the variant where the original grid is kept, and long-range edges added, with a small starting grid. Some minor rewiring of the outer edge of the original grid will be needed to create the drop-off. Similarly, for the Kleinberg model, some of the original grid edges will need dropped (or rewired depending on how the graph is developed).

7.3.2 Reciprocal Attachment

The small-world graph generators we considered both have mechanisms to add long-range connections to the graph. However, they do not reciprocally add connections to other close neighbors, which is needed to generate the "humps" visible in the protein contact distributions. This could be added to the graph construction process after an initial long-range connection has been made by adding connections to sequence neighbors with high probability but dropping off fast enough to accomodate hump sizes/properties. As an analogy, I think of something along the lines of the correlation structure associated with an AR(1) process with high ρ could be used to govern the addition of edges. For example, after adding a random long-range connection, treat that as the midpoint of a new neighborhood connection. Add connections to sequence neighbors who are one away in sequence from each amino acid in the long-range connection with probability ρ, where $\rho \geq .95$ (.95 chosen as an example). Add connections to sequence neighbors who are two away in sequence with probability ρ^2, etc. The distribution used to govern the reciprocal attachments, if it generates graphs that look like the protein graphs, may shed some light on protein packing. The long-range connection distribution also needs adjustment to deal with the "hump" properties of the protein contact distributions.

7.3.3 Long-Range Connections

Adding reciprocal attachment will go a long way towards generating random graphs that behave like protein graphs. However, adjusting the long-range connection distribution to accomodate "hump" properties is a challenge. Unlike the Kleinberg model, where the long-range connection probability is governed by the inverse square of the grid distance between two vertices, long-range connections will likely need to be governed by sequence separation with intermediate distances given the highest probabilities. Then, once an attachment is made, and reciprocal attachments completed, constraints should be made to avoid adding additional edges within those neighborhoods. For example, if a protein has 100 amino acids, it is unlikely amino acid 1 and amino acid 100 are in contact. It is more likely that amino acid 1 contacts amino acid 30, and amino acid 71 contacts amino acid 100. If a long-range connection is added between say, amino acid 25 and amino acid 55, and reciprocal connections are completed, we should not add another long-range connection between amino acid 26 and amino acid 53, because this was already a considered reciprocal connection for a long-range connection. Additional challenges lie in capturing differences by protein class.

8. Discussion, Conclusions, and Future Work

In this paper, we have discussed selected results on protein graph construction mechanisms. We showed that all-atom single contact graphs with no filter can be considered to be small-world. We also examined differences in average degrees among the graph constructions. For predicting folding rate constants, we found that all-atom single contact graphs perform comparably to the originally suggested all-atom multiple contact graphs to form absolute contact order. We took a slightly more in-depth look at the small-world graph properties of average path length and clustering coefficient, before turning to questions about random graphs and generating protein graphs. After supplying evidence that current small-world graph generators do not generate protein-like graphs, we outlined properties needed in a model to succeed in generating protein-like graphs.

As suggested in various sections, particularly in Section 7, much related work remains. In particular, we plan to look at graph properties as measures of protein stability and study relationships to folding/unfolding, such as ERIP (7), but extended to other measures like centrality. Further examinations of centrality measures with different filters applied would also be interesting due to their potential to identify important amino acid contacts for folding. Clearly, there is significant work in determining an appropriate graph generator for protein graphs, and we have active work in this area. We hope this work will shed light on protein folding and amino acid packing properties. We may also investigate whether or not the protein folds may be characterized by their graph properties. The different graph constructions generate different numbers of edges, and we have work examining relationships between the numbers of edges to examine packing properties as well. Finally, there is significant work ahead in obtaining a larger, representative sample of proteins and their graphs from the PDB, even if kinetic/thermodynamic information is not available for those proteins.

References

[1] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, *The protein data bank*, Nucleic Acids Research, 28 (2000), pp. 235–242.

[2] D. Chakrabarti and C. Faloutsos, *Graph mining laws, generators, and algorithms*, ACM Computing Surveys, 938 (2006).

[3] L. Greene and V. Higman, *Uncovering network systems within protein structures*, J. Mol. Biol., 334 (2003), pp. 781–791.

[4] M. Gromiha and S. Selvaraj, *Comparison between long-range interactions and contact order in determining the folding rate of two-state proteins: Appli-*

cation of long-range order to folding rate prediction, J. Mol. Biol., 310 (2001), pp. 27–32.

[5] M. Habibi, C. Eslachi, M. Sadeghi, and H. Pezashk, *The interpretation of protein structures based on graph theory and contact map*, Open Access Bioinformatics, 2 (2010), pp. 127–137.

[6] D. N. Ivankov, S. O. Garbuzynskiy, E. Alm, K. W. Plaxco, D. Baker, and A. V. Finkelstein, *Contact order revisited: Influence of protein size on the folding rate*, Protein Science, 12 (2003), pp. 2057–2062.

[7] J. Jung, J. Lee, and H. Moon, *Topological determinants of protein unfolding rates*, Proteins: Structure, Function, and Bioinformatics, 58 (2005), pp. 389–395.

[8] A. Krishnan, J. Zbilut, M. Tomita, and A. Giuliani, *Proteins as networks: Usefulness of graph theory in protein science*, Current Protein and Peptide Science, 9 (2008), pp. 28–38.

[9] M. Newman, *Networks: An Introduction*, Oxford University Press, New York, 2010.

[10] V. Nguyen and C. Martel, *Analysis and models for small-world graphs*, in Proceedings of Symposium on Discrete Algorithms, vol. 16, ACM-SIAM, 2005.

[11] K. Plaxco, K. Simons, and D. Baker, *Contact order, transition state placement and the refolding rates of single domain proteins*, Journal of Molecular Biology, 277 (1998), pp. 985–994.

[12] M. Rodionov and M. Johnson, *Residue-residue contact substitution probabilities derived from aligned three-dimensional structures and the identification of common folds*, Protein Science, 3 (1994), pp. 2366–2377.

[13] M. Vendruscolo, N. Dokholyan, E. Paci, and M. Karplus, *Small-world view of the amino acids that play a key role in protein folding*, Physical Review E, 65 (2002).

[14] A. Wagaman and S. Jaswal, *Data mining in exploring protein thermodynamics and kinetics relationships*, in JSM 2011 Proceedings, American Statistical Association, 2011, pp. 3157–3165.

[15] D. Watts and S. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.