Graph Construction and Random Graph Generation for Modeling Protein Structures

Amy Wagaman

Department of Mathematics, Amherst College, Amherst, MA 01002, USA

Received 12 October 2012; revised 7 June 2013; accepted 22 July 2013
DOI:10.1002/sam.11203
Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: Researchers often model folded protein structures as graphs with amino acids as the vertices and edges representing contacts between amino acids. The vertices in these graphs are naturally ordered in the amino acid sequence order. There are many different graph construction methods and 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. We propose a new graph property for graphs with ordered vertices, the contact distribution, and propose a method of reciprocal attachment to merge neighborhoods for protein graphs. © 2013 Wiley Periodicals, Inc. Statistical Analysis and Data Mining, 2013

Keywords: 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. There are many instances where 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 [1–4]. Many possible graph constructions exist due to different representations of the protein. There is no consensus about which construction to use in the literature and little discussion as to what the differences and consequences of each construction are. It is also unclear whether current graph generators can generate random graphs that behave similar to protein graphs. We address these issues by comparing graph concepts across various graph constructions based on protein structures and graphs generated by random graph generators. We also examine the distribution of long-range contacts in protein graphs compared to mechanisms for adding long-range contacts in small-world graph generators, and explore whether or not the probability models directing long-range contacts in use are appropriate for protein graphs.

A graph is a collection of vertices and edges (V, E), where an edge is a two-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., the graphs do not have multi-edges (meaning there is either no edge or only one edge between any two vertices), but one protein representation allows multi-edges in the corresponding graph. Protein graph 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 pairwise distance between amino acids is calculated, and if it is below a threshold, the amino acids are in contact. We use Euclidean distance between atoms while the protein is folded as our distance to determine contacts, which is the most common distance used based on our literature survey (results below).

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 side chain which differs between amino acids. In

Correspondence to: A. Wagaman (awagaman@amherst.edu)

the backbone, there are two carbon atoms—commonly called Carbon-Alpha (C-Alpha) and Carbon-Beta (C-Beta), except glycine which has no C-Beta. These atoms are convenient points of reference. When dealing with distances on this atomic scale, the distance unit is the Angstrom (Å). Additionally, 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 using a filter. 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 by other researchers and the results are entered into a freely available database (RCSB) [5]. For our work, we use the protein databank (PDB) files associated with our proteins downloaded from RCSB.

We offer a representative literature review of applications where protein graphs 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 protein representation), 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 (AA) representations. The literature review is summarized in Table 1.

Protein graphs (contact maps) have been used since the 1970s [6]. As seen in Table 1, they continue to be used in current research. The most common representation is C-Alpha. Distance cutoffs for determining contacts typically range from 5–10 Å. Filters are not universally used. In our examples, the filters occurred at different sequence separations. For example, amino acids needed to be more than two amino acids apart for contacts to count in Krishnan's work [3], but Gromiha focused on long-range contacts more than 12 amino acids apart in sequence [8]. AA graphs allow multi-edges unless restricted. Finally, we see from the applications that much of the related work deals with protein folding, but there has been a recent shift toward 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. Also, the question of whether or not small-world graph generators can generate random graphs that mimic protein graphs has not been addressed. 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. First, we introduce relevant graph definitions and a new graph property for protein graphs in Section 2. Our methods for protein graph construction and background on existing methods for random graph generation are presented in Section 3. We introduce our protein dataset in Section 4. In the first part of our results, Section 5, we show the impact of the various construction methods for protein graphs on small-world graph properties. Then in the second part of our results, Section 6, we demonstrate that current random graph generators (using a rewired ring model as a small-world generator) do not yield realistic protein-like graphs, and we propose a reciprocal attachment method for a random protein graph generator. Finally, we conclude with discussion and future work in Section 7.

2. GRAPH CONCEPTS AND A PROPOSED GRAPH PROPERTY

As seen in Section 1, some researchers have computed graph concepts for protein graphs and evaluated their use in understanding protein folding [1–4,10]. We define the graph properties that we examine in this section. Note that many graph concepts do not have adjusted computations for graphs with multi-edges. As a result, our primary focus is to compare the simple graph constructions. For notation, the vertices of our graphs are the amino acids, labeled

Table 1. Example applications of protein graphs in the literature.

First author	Year	Rep.	Dist.	Filter	Application	Citation
Rodionov	1994	C-Beta	—	No	Contact substitution	[6]
Plaxco	1998	All-atom	6 Å	No	Folding rate prediction	[7]
Gromiha	2001	C-Alpha	8 Å	Yes	Folding rate prediction	[8]
Vendruscolo	2002	C-Alpha	8.5 Å	No	Small-world graphs	[1]
Ivankov	2003	All-atom	6 Å	No	Folding rate prediction	[9]
Greene	2003	All-atom	5 Å	Yes	Proteins as graphs	[2]
Jung	2005	C-Alpha	8 Å	No	Unfolding rate prediction	[10]
Krishnan	2008	C-Alpha	6 Å	Yes	Proteins as graphs	[3]
Habibi	2010	C-Alpha	8 Å	No	Proteins as graphs	[4]

Notes: Rep. = protein representation used, Dist. = Euclidean cutoff distance used. Filter simply refers to presence or absence of a filter.

from 1 to n in sequential order, and there are a total of m edges determined by contacts, where m and n depend on the protein and construction methods used. A typical representation of the graph is its adjacency matrix, **A**. The matrix **A** is n by n and the ijth entry in the matrix is the number of edges between vertex i and vertex j. Thus, all entries in **A** will be 0 or 1 for simple graphs. For further details or as an introduction to graphs, see Ref [11]. We begin our definitions with the degree of a vertex, recalling that we are dealing with undirected graphs.

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

For each vertex, the number of edges that connect to that vertex is the degree of the vertex. In notation, this is represented as $q_i, i = 1, \ldots, n$, and is easily computed from the adjacency matrix as $q_i = \sum_{j=1}^{n} A_{ij}$. For protein graphs, the degree 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 we have the constraint $\sum_{q=0}^{n-1} p_q = 1$. It is not uncommon for the degree distribution to follow a power law, such that $p_q = Cq^{-\alpha}, 2 < \alpha < 3$, where C is an appropriate constant [11]. Finally, degree may be used as 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 if centrality is of primary interest [11].

2.2. Graph Concepts and Small-World Properties

The average path length (APL) and the clustering coefficient (CC) of a graph are common and easily computed graph concepts. Shortest path length is the minimum number of edges that must be traversed to go between two vertices. APL is the average of all the shortest path lengths when considering all pairs of vertices [11]. In general, the CC is a measure of how tightly clustered the graph is. There are several nonequivalent definitions of the CC. By the first definition, it is computed as six times the number of triangles divided by the number of paths of length two in the graph [11]. In other words, the CC is the number of triangles out of the number of possible triangles starting from two connected sides out of all triples of vertices. A second definition of the CC allows us to possibly identify important individual vertices because under this definition, the CC is computed first for each vertex, and then averaged across all vertices. For each vertex, v, the local CC 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, but localized to each vertex), with vertices ignored which have fewer than two neighbors. Then, the average is taken over all vertices that have at least two neighbors [11]. We obtain the average local CC over all vertices with at least two neighbors and use it as the CC for the graph. As we stated, these two definitions are not equivalent, and we use the latter local approach.

Many social and biological networks are much more highly clustered than random graphs with similar numbers of vertices. Graphs with high CCs compared to random graphs of the same size and fairly low values for APL are often termed small-world graphs [12]. Researchers have shown protein graphs exhibit small-world tendencies [1,2]. Small-world graphs and their properties have been studied in a variety of applications [12–15].

2.3. New Graph Property: Contact Distribution

Contact distribution is a new graph property that we propose for protein graphs due to the natural ordering of vertices. We define it generally for simple graphs only (not multi-edges) where the vertices are ordered. First, we set edge weights for the graph to be equal to the 'sequence' separation of the connected vertices (i.e., amino acids) [easily computed by subtracting their vertex numbers (larger−smaller)]. Then the contact distribution is the distribution of the weights along the edges in the graph. For protein graphs, it is the distribution of sequence separation for amino acids in contact. Contact distribution allows us to study how the edges are distributed across the possible vertex separations based on the vertex ordering. Let r_i be the fraction of contacts or edges that occur at a sequence separation of i (out of the $n-i$ possible edges at that separation distance). Each r_i is between 0 and 1, and a simple rescaling $s_i = r_i\left(\frac{n-i}{m}\right)$ 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, and our usual choice for our plots of the contact distribution.

Briefly, we highlight the key difference between degree distributions and contact distributions. For a degree distribution, each vertex is assigned a degree, and we look at the distribution of those values. For a contact distribution, each edge, not vertex, is assigned a separation value, based on the ordering of the vertices, and we look at the distribution of the separation values. Degree distributions can be used to generate random graphs with the same distribution [11]. A similar method should be possible for a contact distribution (ignoring any other graph properties). For example, if you need three edges with a separation of 12, then draw three edges at random from the list of pairs of vertices with separation 12, and add them to the graph, and repeat this process for every separation distance. Additional challenges presented in later sections mean this simple procedure will

not help generate realistic protein graphs without other considerations.

3. METHODS FOR PROTEIN GRAPH CONSTRUCTION AND RANDOM GRAPH GENERATION

3.1. Protein Graph Construction Methods

We examine several aspects of graph construction: the protein representation (atoms used to determine distance), the distance cutoff, and filters, which are used to eliminate contacts.

3.1.1. 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 representation as CA (C-Alpha).

The second is an AA representation where all atoms, except hydrogen, 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, which is used to determine if edges are present. Thus, this graph only has single edges. This representation is referred to as AA, or AA single contact. Hydrogen 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 (S. Jaswal, personal communication, 2011).

Finally, we use the same AA 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 graph construction, multi-edges may result between amino acids, so we refer to the representation as MC (multiple contacts).

3.1.2. Distance

We examined distance cutoffs from 6 to 12 Å in steps of 0.5 Å. We did in-depth examinations of graph concepts at 6, 8, and 10 Å, though the patterns we found are similar for each distance and we focus on 8 Å in the discussion.

3.1.3. Filter

Filters are designed to remove contacts from the graph. Those contacts may be trivial or nontrivial. 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 our filter to be indexed by a parameter k. $k = 1$ means the main diagonal is removed, so this is equivalent to the original adjacency matrix. $k = 2$ means that the first diagonal and main diagonal are removed, meaning that both have all values set to 0. Higher values of k remove additional diagonals as well as all previously removed diagonals.

We note an important cutoff for choices 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 to 20 across our different distances and representations, though at times we focus on filters of $k = 1$, 4, and 10. We chose these main filters to compare the original graph ($k = 1$), a graph with trivial contacts removed ($k = 4$) but where alpha-helical contacts remain, and a graph where only long-range contacts remained ($k = 10$). To give a better picture of the changes due to filters we also have results for $k = 2$, 6, and 8, in our results tables for APL and CC.

3.2. Methods for Random Graph Generation

The most basic random graph model is the Erdös-Rényi random graph or ER model. The ER model can be described in terms of n vertices and either having m edges or defining p as the probability of an edge between any two vertices. The ER model is often referred to as the Poisson model, because in the limit of large n, the degree distribution that results is Poisson [11]. 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 [11]. These models however, do not have high CCs, which often occur in real-world graphs. Specific small-world models were developed to achieve the small-world properties.

Generating a small-world graph can be done in several ways. In the 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 rewired (probability w). Using this generation mechanism, long-range connections are introduced, which decreases the APL. However, the CC remains strong due to the starting neighbor connections [12]. We refer to this graph generating model as the rewired ring model. A variant of the proposal keeps all the original connections and adds a few random long-range ones with probability w.

Other models for generating small-world graphs exist. Nguyen and Martel describe Kleinberg's model [15] as well as a generalization [13]. 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, t long-range connections are added based on a probability that is inverse squarely proportional to the grid distance between each pair of vertices [15]. Generalizations are made to models that start with a grid and add t long-range edges under other probability distributions (that can be vertex specific) [13]. Many other random graph models exist, including models for directed graphs, growing graphs, and so on. For a broad review of graph generators, see Ref [16].

4. PROTEIN DATA

Our data consist of 127 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 Ref [17]. For the analysis in this article, the PDBs of the proteins were downloaded from RCSB [5] and processed using a Perl script to obtain the protein graphs under the methods described in the previous section. The graph concepts were then computed from the protein graphs using R [18], the igraph package [19], 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, which we wanted to have for other analyses.

To get a sense of the data, we provide 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 proteins are multistate folders, and 65 are two-state folders. We have folding rate constants for 115 proteins and unfolding rate constants for 49 proteins. The average helical content of a protein in the dataset is 22.48% (median 16%) and average beta-sheet content is 23.87% (median 26%). Finally, we have all four structural classes represented: 28 are class α, 36 are class $\alpha + \beta$, 8 are class $\alpha \backslash \beta$, 48 are class β, and 7 have unknown class (or are fragments).

5. RESULTS COMPARING PROTEIN GRAPH CONSTRUCTIONS ON SMALL-WORLD CHARACTERISTICS

Several researchers have identified protein graphs as small-world graphs [1,2]. We examine the effect of the various graph constructions via the different protein representations, distance cutoffs, and filters on the graphs in a small-world context. We consider APL and the CC individually, and then look at them together from the small-world viewpoint.

5.1. Average Path Length

APL measures the number of edges that must be traversed to move from one vertex to another. It is no secret that the amino acids in proteins are packed together very tightly and so we expect small APLs for the protein graphs, especially at larger distance cutoffs. The results of comparing APL across the various constructions are intuitive for the different representations and filters. The average APLs of protein graphs (and standard deviations in parentheses) are shown in Table 2 and histograms of the distributions at each representation and main filter combination with distance fixed at 8 Å are shown in Fig. 1.

Our analysis shows that as expected, CA graphs have longer APLs than AA graphs at the same filter and distances. Longer distances mean shorter APLs, and higher filters mean longer APLs. Also, generally, the AA graph with a $k = 10$ filter has a shorter APL than the CA graph with no filter, $k = 1$. Our histograms (in Fig. 1) show the distributions of APL at 8 Å, which we see are fairly concentrated at values between 2 and 4. At 10 Å, most of the path lengths for AA or CA graphs are between 1 and 4, but the CA APLs rise toward 6 when distance is set at 6 Å. Yet,

Table 2. Averages (with standard deviations in parentheses) of the average path lengths for our 127 proteins under different representations, distances, and filters.

Rep.	AA			CA		
Filter\Dist.	6 Å	8 Å	10 Å	6 Å	8 Å	10 Å
$k = 1$	2.81 (0.83)	2.20 (0.58)	1.91 (0.48)	5.38 (1.97)	3.42 (1.06)	2.54 (0.71)
$k = 2$	2.85 (0.83)	2.22 (0.57)	1.93 (0.47)	5.93 (2.15)	3.47 (1.06)	2.56 (0.71)
$k = 4$	3.06 (0.85)	2.31 (0.57)	1.98 (0.45)	5.64 (2.77)	4.06 (1.26)	2.70 (0.72)
$k = 6$	3.26 (0.95)	2.38 (0.57)	2.03 (0.45)	5.44 (2.61)*	4.18 (1.54)*	2.85 (0.74)
$k = 8$	3.35 (0.96)*	2.47 (0.59)	2.09 (0.45)	5.45 (2.79)*	4.18 (1.54)*	2.94 (0.79)
$k = 10$	3.42 (0.97)*	2.54 (0.59)*	2.13 (0.45)	5.43 (3.21)*	4.27 (1.57)*	3.01 (0.81)*

Notes: An * indicates several graphs were not connected at that filter/distance/representation combination so APL was not computed for few graphs (max number of NAs = 3).

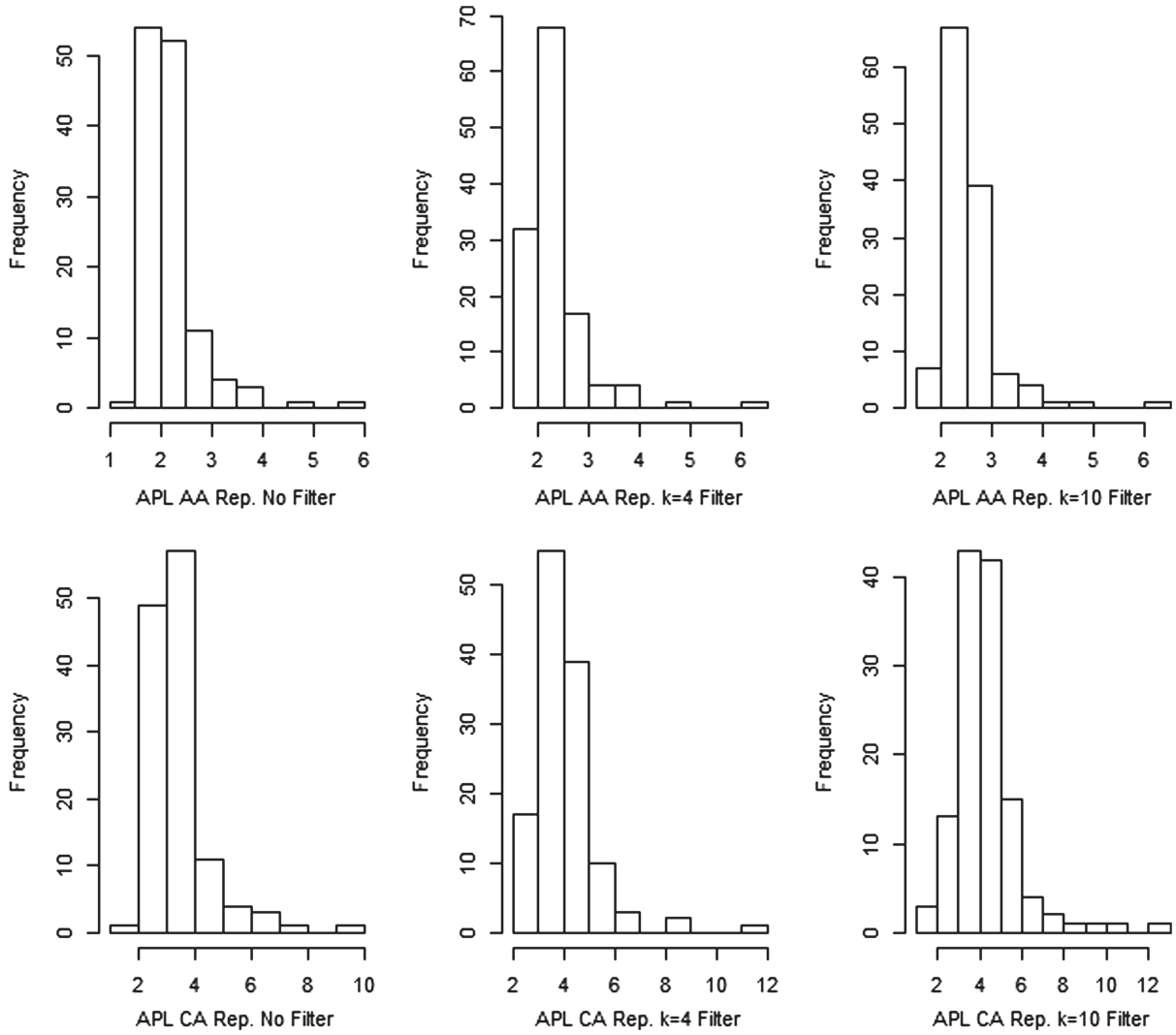

Fig. 1 Histograms of average path length for the 127 proteins in our dataset at each representation and main filter level ($k = 1$, 4, or 10) with distance fixed at 8 Å.

even when we look at 6 Å for CA graphs, the longest path lengths are between 10 and 15, and most are between four and seven. Considering the size of some of these graphs, that is impressive, it appears that we can transverse the graph using very few vertices.

Briefly, we consider the relationship between APL and graph size focusing on differences between AA and CA representations at 8 Å. Figure 2 is a scatterplot showing the relationship with no filter applied. APL does increase slightly as graph size increases, as expected. For the AA representation, applying a filter does not increase the APL much. Comparing the main filters, at 8 Å, the average increase at $k = 4$ is only 0.11 and at $k = 10$ this goes up slightly to 0.34 from the $k = 1$ reference point, but is still less than one additional edge, and is similar for the other distances we examined. CA graphs have larger increases in APL as filters are applied. In CA graphs, for the main filters, at 8 Å, for $k = 4$, APLs are on average, 0.64 longer than their $k = 1$ counterparts, and $k = 10$ APLs are 0.85 longer on average compared to $k = 1$. 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 representation. The average difference between AA and CA APLs for $k = 1$ at 10 Å is 0.63, at 8 Å is 1.22, and at 6 Å is 2.57. Thus, at higher distances, the representation difference is not as pronounced.

5.2. Clustering Coefficient

Comparing the constructions in terms of their CCs is also fairly intuitive. For the protein graphs, we display the average CCs (and standard deviations in parentheses)

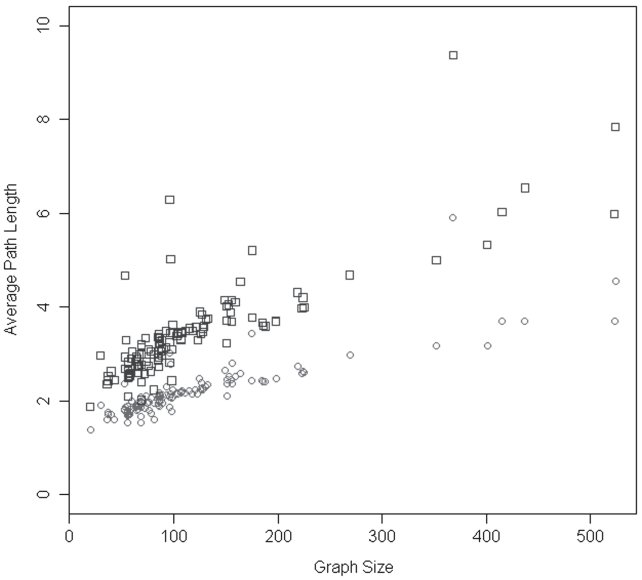

Fig. 2 Graph size versus average path length of each protein in the dataset at 8 Å with no filter $k = 1$. CA average path lengths are blue squares, while AA average path lengths are red circles. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

in Table 3 and histograms of the distributions at each representation and main filter combination with distance fixed at 8 Å shown in Fig. 3.

Applying a filter reduces the CC significantly. Our results show that notably, even at the greatest distance we have of 10 Å, some of the proteins have a zero for their CC under a CA construction. This occurs even more at 6 Å, eventually reaching a point where some proteins have no vertices with even two connected neighbors. CCs increase as distance increases, and the representations do not result in terribly different CCs if no filter is present ($k = 1$), though the AA values are a little higher than the CA values. Once filters are applied, AA graphs have much higher CCs than CA graphs, and this is more pronounced at lower distances. The significant drop in the CC when a filter is applied has some implications for using the CC as a measure of how tightly clustered protein graphs are, at least when filters are applied. For long-range filters, say $k = 8$ or 10, if using the CA representation, the CCs drop to near zero (as evidenced in the histograms, even at 8 Å) and there is not much variability in their values. Hence, it might be best to only consider the CC without filters applied, or to develop a new way to quantify long-range neighbor relationships.

Finally, we consider the relationship between graph size and the CC. For the ER model, the CC is equal to the probability of any two nodes being connected, or equivalently, the average degree of the graph divided by the graph size. Thus, for random graphs, a log−log plot of the ratio of the CC to average degree versus graph size should align along a straight line with slope equal to negative one [20]. However, as in Ref [20] with a log−log plot, most real networks have CCs that appear to be independent of graph size. To examine the relationship between graph size and the CC for our nonfiltered protein graphs, we make a log−log plot similar to the one in Ref [20] for our protein data (shown in Fig. 4) with distance set at 8 Å. For graph sizes over 100, it appears that the CC is independent of the graph size. For graph sizes smaller than 100, we see the CC divided by average degree ratio increases as the graph size decreases. This is not unexpected, as in proteins with a small number of amino acids, the backbone connections via sequence neighbors make up the bulk of the contacts that are present, and these connections are often connected triples and hence increase the CC. The average degree for protein graphs with fewer than 100 amino acids is also less than the average degree for graphs with greater than 100 amino acids (20.29 vs. 23.15 for AA method and 9.48 vs. 9.95 for CA method at 8 Å), which also contributes to an increased ratio of CC to average degree. Hence, for large proteins, we see the nonfiltered protein graph CCs appear to be independent of protein size, but for small proteins, we see some dependence on protein topology, particularly the

Table 3. Averages (with standard deviations in parentheses) of the clustering coefficients (CC) for our 127 proteins under different representations, distances, and filters.

Rep.	AA			CA		
Filter\Dist.	6 Å	8 Å	10 Å	6 Å	8 Å	10 Å
$k = 1$	0.57 (0.04)	0.63 (0.04)	0.68 (0.05)	0.53 (0.05)	0.59 (0.03)	0.62 (0.04)
$k = 2$	0.39 (0.04)	0.51 (0.04)	0.59 (0.04)	0.11 (0.07)	0.33 (0.04)	0.45 (0.04)
$k = 4$	0.21 (0.06)	0.36 (0.05)	0.47 (0.05)	0.03 (0.04)	0.11 (0.07)	0.27 (0.06)
$k = 6$	0.17 (0.05)*	0.30 (0.05)	0.39 (0.06)	0.01 (0.02)*	0.07 (0.05)*	0.20 (0.07)
$k = 8$	0.15 (0.05)*	0.25 (0.06)	0.34 (0.06)	0.01 (0.02)*	0.06 (0.05)*	0.17 (0.07)*
$k = 10$	0.13 (0.05)*	0.22 (0.07)*	0.29 (0.08)	0.01 (0.02)*	0.05 (0.04)*	0.15 (0.07)*

Notes: An * indicates several graphs had no vertices with a minimum of two connected neighbors at that filter/distance/representation combination so CC was not computed for a few graphs (max number of NAs = 7).

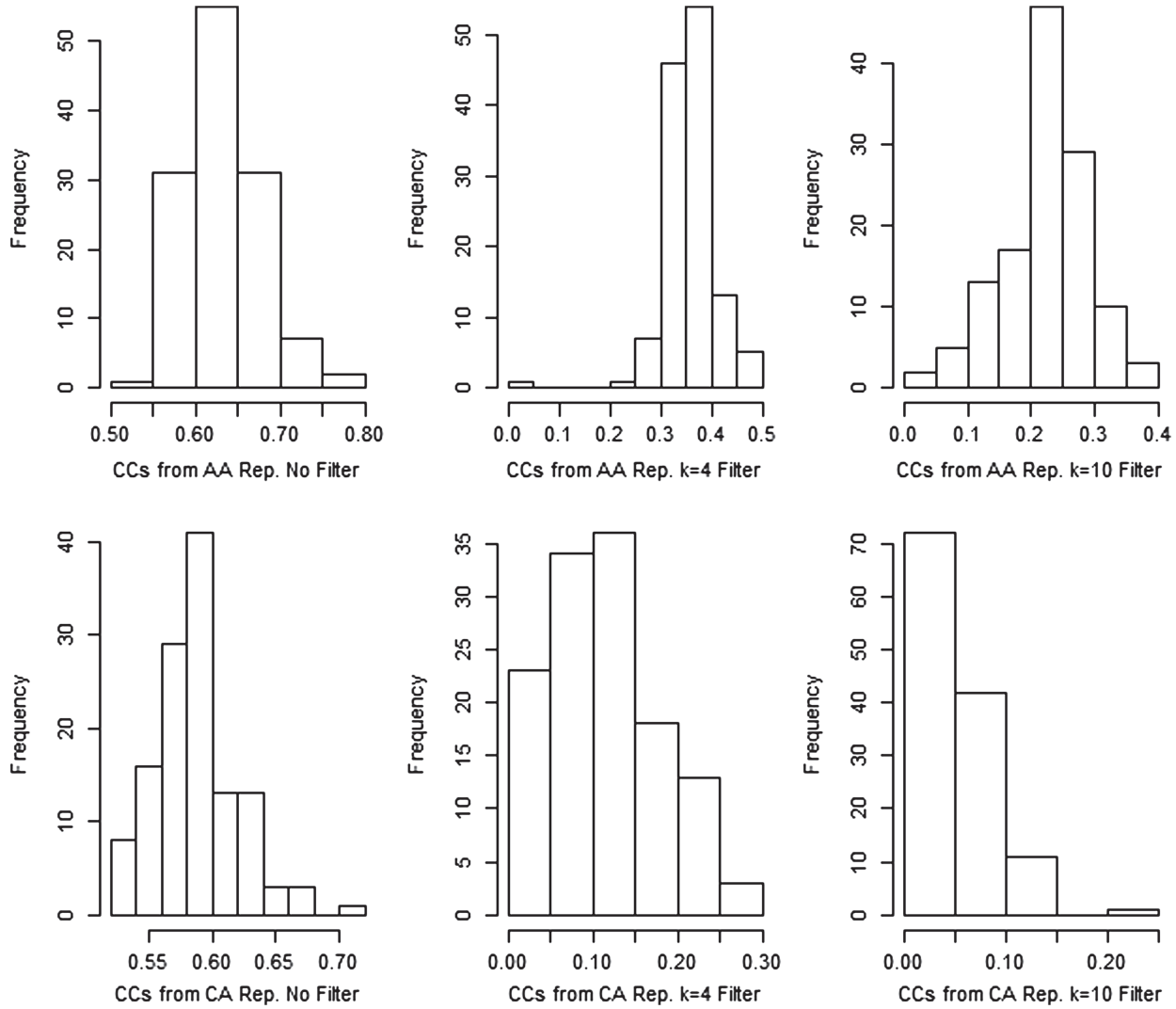

Fig. 3 Histograms of clustering coefficients for the 127 proteins in our dataset at each representation and main filter level ($k = 1$, 4, or 10) with distance fixed at 8 Å.

protein backbone that contributes many connected triples of amino acids to the CC.

5.3. Small-World Properties—APL versus CC

Next we consider the interaction of APL and CC, in terms of small-world graphs. Recall that small-world graphs have high CCs and fairly low APLs. We found that the choice of distance cutoff does not influence the overall pattern in our results, so we fix it here at 8 Å. Figure 5 shows the average CCs plotted against the APL for our data under both the AA and CA representations and at the main filters of $k = 1$, $k = 4$, and $k = 10$. Clearly, the nonfiltered graphs ($k = 1$; at the far right in Fig. 5) are the ones that meet our criteria for small-world graphs. Also, at each main filter level, the AA construction appears to have lower APLs and slightly higher CCs than the CA representation, which is expected because the AA graph contains more edges with more contacts. At $k = 4$ (triangles in Fig. 5), it is clear the CA graphs are no longer small-world (their CCs are too low), but many of the AA graphs still have APLs from two to four and CCs in the range of 0.30–0.45, and this may still fit the small-world criteria. These CC values for the AA graphs at $k = 4$ are a little larger than we would expect for a random graphs of these sizes, but not by very much, because the mean degree to graph size ratio (≈CC) for the 127 protein graphs on average is 0.24 with a standard deviation of 0.10 [11,12]. By $k = 10$, it is clear that neither the AA or CA graphs could be termed small-world. The APLs rise and the CCs plummet (near zero for CA and 0.05–0.25 for AA). In summary, to pick a construction that most clearly satisfies the small-world conditions, one must use the nonfiltered

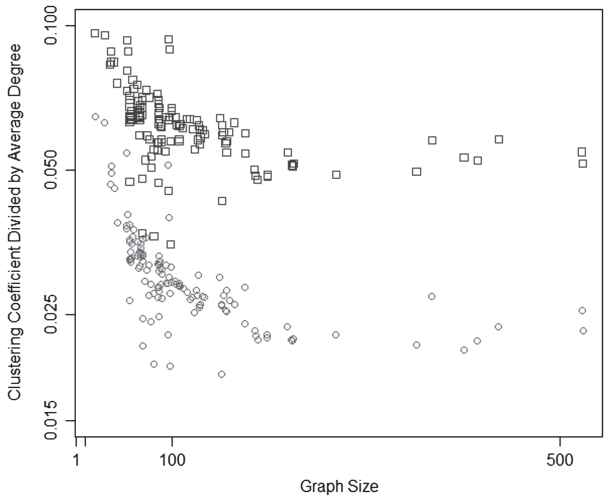

Fig. 4 Graph size versus clustering coefficient divided by average degree of each protein in the dataset at 8 Å with no filter $k = 1$ on a logged scale (both axes). Results from the CA method are blue squares, while results from the AA method are red circles. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

graph from an AA representation at a reasonable distance. We note that our CCs (shown in Fig. 5) are higher than those in Ref [2], but their graphs were constructed with a distance cutoff of 5 Å, compared to ours at 8 Å.

It is clear from Fig. 5 that the application of filters destroys the small-world property because the CC plummets. This is intuitive because the natural ordering of amino acids and methods of graph construction result in sequence neighbors being graph neighbors, and the CC should drop 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 representation. One can also note that for the CA representation and $k = 10$ filter, a group of proteins has zero as the value of the CC, so that effectively all connected vertex triples have been dismantled.

6. RESULTS ON CONSTRUCTING A RANDOM PROTEIN GRAPH GENERATOR

In this section, we investigate whether or not small-world graph generators can generate graphs that mimic protein graphs. The main feature we try to mimic is the newly proposed contact distribution of the protein graphs. 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. Additionally, the graph generator mechanism may be useful for modeling other real-world networks. However, existing

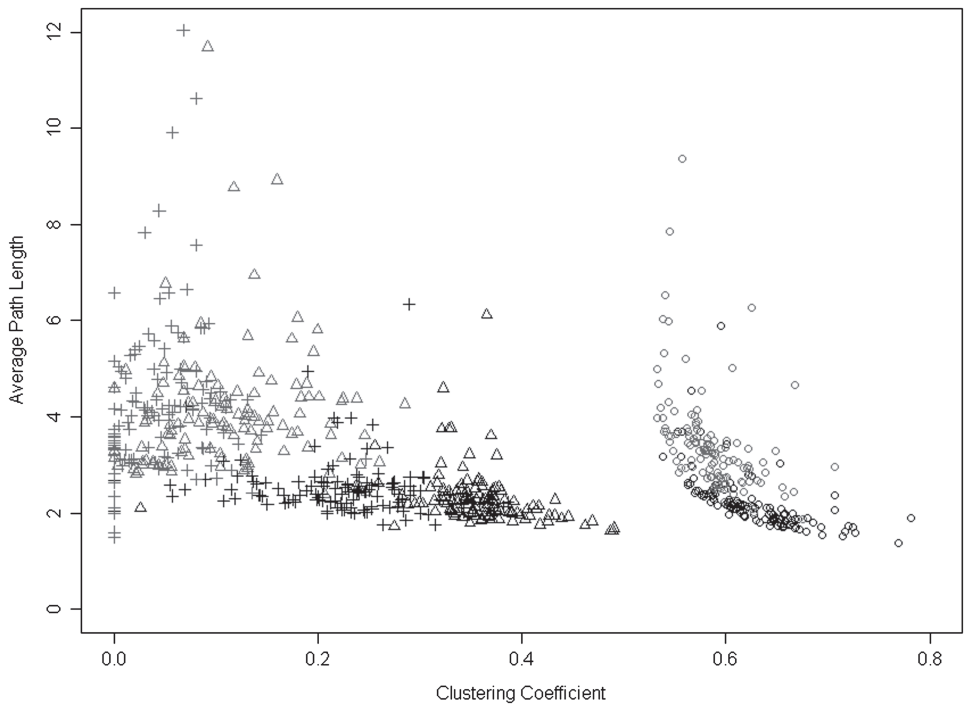

Fig. 5 Scatterplot of average clustering coefficient versus average path length for the AA (black) and CA (red) representations at three different filter levels at 8 Å. Circles are $k = 1$ filter (i.e., nonfiltered graph), triangles are $k = 4$, and plus signs are $k = 10$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

mechanisms in small-world graph generation for adding long-range contacts do not appear to generate realistic protein graphs; so we discuss a reciprocal attachment method in conjunction with other changes to graph generating procedures (including a different probability distribution to govern long-range contacts) that may help to generate realistic protein graphs.

6.1. Random Graphs Versus Protein Graphs

Briefly, we explore the differences between randomly generated small-world graphs and protein graphs with no filter via their contact distributions as well as APLs and CCs (as example graph properties). We do not compare the filtered protein graphs with existing graph generators and leave this as an area for future work, because we are interested in the small-world aspects of proteins seen by other researchers.

6.1.1. Protein contact distributions

Beyond being small-world in nature, another feature that makes protein graphs interesting is the natural order of the vertices, and its consequences. Again, contact distribution is the distribution of sequence separation values for vertices in contact and may be scaled in one of two ways—either consider the number of edges at each sequence separation i as a fraction of the maximum possible at each sequence separation $(n - i)$, or as a fraction of existing edges (our preference). Contact distributions for protein graphs have interesting shapes due to protein folding patterns. An example contact distribution under the AA construction at 8 Å for PDB 1APS with no filter is shown in Fig. 6. The graph has 98 vertices and 1658 edges. The rescaling was chosen as a fraction of existing edges. The effect of a filter k on a contact distribution is just to set the first k-1 sequence separation proportions to zero, with rescaling as needed to keep any desired constraints.

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 Fig. 6 occurs around sequence separation 30. This might be because amino acid 12 was in contact with amino acid 42, which suggests amino acid 11 might be in contact with amino acids 42 or 43, and that amino acid 12 might be in contact with amino acid 43, and so on. 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, taking the vertex order to be as provided by the generating algorithm.

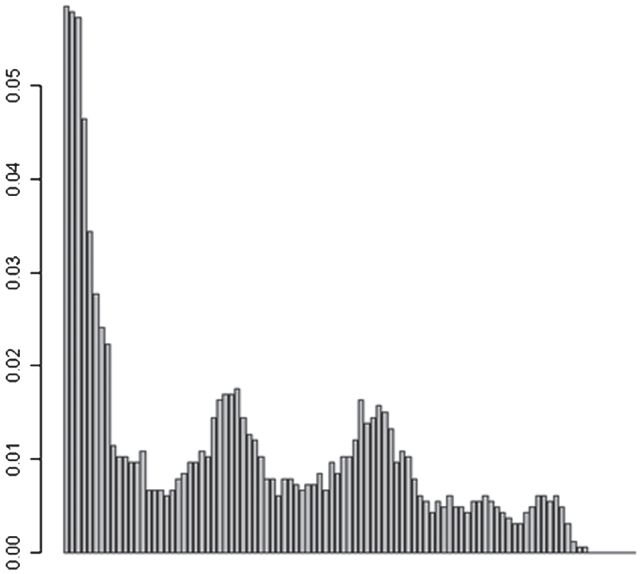

Fig. 6 Contact distribution of PDB 1APS under AA construction at 8 Å with no filter ($k = 1$).

An example contact distribution from a rewired ring model (igraph function watts.strogatz.game(1, 100, 16, 0.3) [19]) with 100 vertices, 1600 edges, and a rewiring probability of 0.3 is shown in Fig. 7. The number of vertices and edges were chosen to be similar to the graph of protein PDB 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 rewired ring model graph does not look at all like the contact distribution of the protein graph.

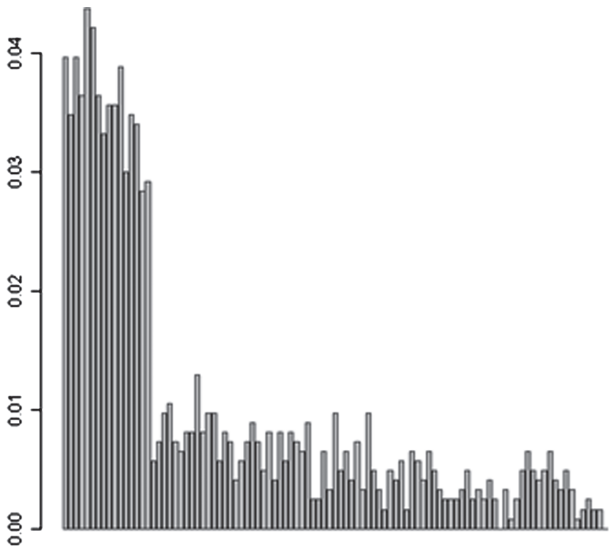

Fig. 7 Contact distribution of a small-world graph generated from a rewired ring model (100 vertices, 1600 edges, rewire probability = 0.3).

Currently, we are investigating automatic ways of quantifying the differences in the distributions. 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, CCs, and APLs may be similar. This leads to some natural questions. How can we obtain random protein-like graphs? Can we put protein graphs in a framework where they are a subset of small-world graphs (but with ordered vertices)? What do we learn about protein folding or packing from our adventures in making random protein graphs?

6.1.2. APL and CC from protein graphs versus small-world rewired ring model graphs

To look at differences between the protein graphs and small-world graphs generated via the rewired ring model [12], we examined the distributions of graph size and number of edges at 8 Å for our set of protein graphs under the AA construction with no filter in order to generate similar sized graphs with the rewired ring model generating algorithm using igraph [19]. The graph size for our protein graphs seems to be fairly well-modeled by a gamma distribution with shape 2.7944 and rate 0.0238. We sampled size values from that distribution to set sizes for the random graphs. We did implement a constraint that sizes could not be smaller than 20, and redrew values if this occurred. Twenty amino acids is the minimum size for a protein in our dataset. After the sizes were set, we drew a value for the ratio of edges to graph size from two different uniform distributions [one for graphs with more than 199 vertices, a Uniform on (11, 14), and one for those graphs with fewer than 200 vertices, a Uniform on (9, 12)], chosen based on the distributions of these ratios for our protein graphs, at a distance of 8 Å. We sampled the ratio after the size selection to appropriately model the differences in the distribution of number of edges based on size. We model the ratio, instead of the number of edges, because that makes it a more natural input to the rewired ring model in igraph, where we need to select a value of the number of neighbors to be connected at the initial ring construction. The initial connections then have a probability of being rewired, and we simplified (removed multi-edges from) any graphs that ended up with multi-edges after rewiring. After some trial and error on a small grid of shape values for a beta distribution to model the rewire probability, to see if we could get the APL and CC values to be near protein graph values, we decided to model the rewire probability as a beta distribution (3, 10). We then generated 10 000 rewired ring small-world models according to these various inputs, and compared the results in terms of sizes (number of vertices), number of edges, APLs, and CCs to our AA protein graphs at 8 Å with no filter. The results are shown in Table 4.

Table 4. Average (with standard deviations in parentheses) graph sizes, number of edges, average path lengths, and clustering coefficients of 10 000 randomly generated rewired ring small-world models (parameters specified in text) compared to 127 protein graphs from the AA representation with no filter at 8 Å.

Measure	Rewired ring model graphs	Protein graphs
Graph Size	118.02 (69.68)	117.46 (90.42)
Number of edges (overall)	1300.70 (881.40)	1324.34 (1144.55)
Number of edges (size $>$ 199)	3200.87 (705.44)	4247.67 (1327.19)
Number of edges (size $<$ 200)	1035.19 (490.78)	1019.30 (533.39)
APL (overall)	1.81 (0.25)	2.20 (0.58)
APL (size $>$199)	2.09 (0.14)	3.45 (0.97)
APL (size $<$ 200)	1.77 (0.24)	2.07 (0.32)
CC (overall)	0.36 (0.17)	0.63 (0.04)
CC (size $>$ 199)	0.25 (0.12)	0.56 (0.01)
CC (size $<$ 200)	0.37 (0.17)	0.64 (0.04)

Table 4 shows that the randomly generated graphs are most similar in size and number of edges to the protein graphs for smaller graph sizes (fewer than 200 vertices), though the overall size and edge values are not far off due to the low number of larger graphs for the graph sizes and number of edges. However, the similarities stop there. Even though we searched for a good rewiring probability, the APLs of the randomly generated graphs are too short compared to the protein graphs. The CCs have a similar problem. The CCs from the randomly generated graphs are very low compared to the protein graphs. Thus, it appears that while we can mimic the properties of size and number of edges, the rewiring process with a fixed rewire probability from a starting ring model does not result in a graph that behaves like a nonfiltered protein graph in terms of APL or CC. This is not too surprising, but it means a more appropriate model is needed. In our next section, we consider what steps and/or properties are needed to generate small-world graphs that behave like protein graphs.

6.2. Considerations for Constructing Random Protein-Like Graphs

6.2.1. 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, Fig. 6, 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 rewired ring or Kleinberg graph model [12,15]. The rewired ring 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, meaning the rewire probability will not be constant for all edges. Similarly, for the Kleinberg model, some of the original grid edges will need dropped (or rewired depending on how the graph is developed), and this may be done with probabilities based on their amino acid chain separation distance.

6.2.2. *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 graph neighbors with high probability, but dropping off fast enough to accommodate hump sizes or properties. As an analogy, something along the lines of the correlation structure associated with an AR(1) process with high p 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 vertex neighbors who are one edge away from the amino acid in the long-range connection with probability p, where $p \geq 0.95$ (0.95 chosen as an example). Add connections to vertex neighbors that are two edges away with probability p^2, and so on. 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.

Reciprocal attachment may be a useful concept for social network models as well. Assume we are modeling an online social network from a school, and each vertex is a student. If a friend of yours makes a new friend connection, you may see that connection, and also wish to connect to that individual, but your desire for the new connection is probably related to how close you are to the friend who made the connection and to the possible new contact. Reciprocal attachment as described here is different from preferential attachment [11]. The network is not growing as it does in many preferential attachment models, instead we are assuming the vertices are already fixed and we want to distribute the edges of the graph appropriately, merging neighborhoods as we add edges.

In the discussion of reciprocal attachment, we did not restrict the attachments to be between near sequence neighbors, but rather to current graph vertex neighbors. The reason is easily illustrated with an example. As you add long-range connections, reciprocal attachments happen based on the current graph. For example, suppose you already added a long-range connection between vertices 5 and 23, and had several reciprocal connections, and a new long-range connection is determined between vertices 23 and 47. You do not want to be limited in reciprocal attachments between vertices near in sequence to 23 and 47. Instead, because vertex 5 was connected to vertex 23, which is now connected to vertex 47, a reciprocal connection between 5 and 47 should be considered (currently 2 edges are needed to get from 5 to 47 based on the current graph, a connection would form a triangle), just like a connection between 22 and 47 (also two edges needed to get from 22 to 47 where a connection would form a triangle). In order to deal appropriately with reciprocal attachments then, we appear to need an updatable adjacency matrix for the graph with path length information or just current path lengths in a matrix (updated after each long-range connection is added), and we do not want our neighborhoods for reciprocal attachments to be constrained to be sequence neighbors. Indeed, the concept of sequence neighbors may be unique to protein graphs, and we need to consider a broader idea to merge the graph neighborhoods after choosing a long-range contact to add.

6.2.3. *Long-range connections*

Developing a reciprocal attachment model will help generate random graphs that behave like protein graphs. However, adjusting the long-range connection distribution to accommodate hump properties is also a challenge. Unlike the Kleinberg model [15], where the long-range connection probability is governed by the inverse square of the grid distance between two vertices, and unlike the variant of the rewired ring model [12] where each long-range connection could be made with equal probability, long-range connections will need to be governed by sequence separation (along a chain, not a two or three-dimensional grid) with intermediate sequence separation 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 acids 1 and 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, 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.

One way to keep track of long-range connections and reciprocal connections already considered would be to use a matrix containing contact probabilities for future long-distance contacts, a long-distance contact probability matrix. For example, to begin, we would fill an n by n matrix with the probability of long-range connections between vertex i and j, with the probability as the ijth entry in the matrix. We would have to define long-range, and place appropriate constraints on the matrix (we may want row sums to be one, so that we actually have a probability distribution for each vertex). Then, we randomly draw a vertex to add a connection to, and use its long-range connection probability distribution (from that row of the matrix) to randomly choose a long-range connection to add. Next, we consider reciprocal attachments related to that long-range connection, using path lengths and whatever probability distribution we have settled on, updating the adjacency matrix and path length information as we go. Before determining the next long-range connection, we update the long-distance contact probability matrix, setting contacts already considered to have future probability 0, and if necessary, rescaling along the rows (and\or columns) to keep desired constraints. We note that this matrix is not a standard contact map (from the protein literature), which contains probabilities of vertices being in contact in the folded state of the protein. Instead, our proposal is an updatable long-range connection probability matrix, and we use it to determine what long-range connections to add, and deal with the short-range connections during the reciprocal steps. Additionally, the stopping point (i.e., how many long-range contacts to add) still needs to be determined and is not derived from the matrix (though you may consider drawing long-range connections until the matrix is full of 0s, as one stopping point).

6.2.4. Differences by protein class

Another challenge with the long-range connections (and even, the starting ring or grid and drop off pattern) is that the pattern in the contact distributions of proteins appears to differ by protein structural class. Further analysis is still needed on these possible differences, but if confirmed in future work, we may use different distributions to govern long-range connections based on protein structural class. We still aim for a general model first.

6.2.5. Tying in amino acid information

A major feature related to proteins that is left out of our graphs is the amino acid type. There are only 20 amino acids involved, and a 20 by 20 matrix of contact types can be constructed for any protein. However, incorporating this information into our graphs is a challenge. Approached differently, from the realm of contact maps where contact predictions are desired, using the amino acid information and contact probabilities based on type and sequence separation is important. Models to predict contacts have become quite sophisticated, but success rates still vary and techniques may work well for some proteins and not others (especially if they are dependent on template proteins or protein fragments). For now, we are not concerned with adding amino acid information, and want to pursue work on reciprocal attachment models, because they have broader applications than just to generate graphs that behave like protein graphs. As mentioned above, reciprocal attachment may be useful for modeling social networks, where a friend request may result in a short series of reciprocal requests from a network of friends.

7. DISCUSSION, CONCLUSIONS, AND FUTURE WORK

In this article, we have discussed selected results on protein graph construction mechanisms and challenges in generating random graphs that behave like protein graphs. We found that the sparsity of Carbon-alpha graphs compared to AA single contact graphs leads the Carbon-alpha graphs to have longer APLs and smaller CCs than the AA graphs at every distance and filter. The AA single contact graph with no filter was demonstrated to be small-world. Indeed, introducing any filter destroyed the small-world properties of the protein graphs because the CC dropped while APL was not strongly affected.

Next, we turned to questions about random graphs and generating protein graphs, and introduced a new graph concept—the contact distribution, for use when the vertices of a graph are ordered. After supplying evidence using the rewired ring small-world model, arguing that it does not generate protein-like graphs, we outlined properties needed in a model to succeed in generating protein-like graphs. This included a different long-range contact distribution (yet to be identified) and development of a reciprocal attachment model to merge the neighborhoods brought together by the long-range contact. Research work to compare filtered protein graphs to existing graph generators is also a next step. This will shed light on contacts made by the amino acids in the protein not related to the amino acid backbone.

As suggested in the various sections, particularly in Section 6, 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 [10], 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 developing an appropriate graph generator for protein graphs, and we have active work in this area, pursing reciprocal attachment models in a protein setting with extensions to social network settings (though the vertices are not ordered in that application). We hope this work will shed light on protein folding and amino acid packing properties. We may also investigate whether or not the different 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, to use for all these analyses and graph generator development, even if kinetic or thermodynamic information is not available for those proteins.

ACKNOWLEDGMENT

The author would like to thank an anonymous reviewer for helpful comments used in revision of this work.

REFERENCES

[1] M. Vendruscolo, N. Dokholyan, E. Paci, and M. Karplus, Small-world view of the amino acids that play a key role in protein folding, *Phys Rev E* 65 (2002), 061910.

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

[3] A. Krishnan, J. Zbilut, M. Tomita, and A. Giuliani, Proteins as networks: usefulness of graph theory in protein science, Curr Protein Pept Sci 9 (2008), 28–38.

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

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

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

[7] K. Plaxco, K. Simons, and D. Baker, Contact order, transition state placement and the refolding rates of single domain proteins, J Mol Biol 277 (1998), 985–994.

[8] M. Gromiha, and S. Selvaraj, Comparison between long-range interactions and contact order in determining the folding rate of two-state proteins: application of long-range order to folding rate prediction, J Mol Biol, 310 (2001), 27–32.

[9] 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 Sci 12 (2003), 2057–2062.

[10] J. Jung, J. Lee, and H. Moon, Topological determinants of protein unfolding rates, Proteins: Struct Funct Bioinf 58 (2005), 389–395.

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

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

[13] V. Nguyen, and C. Martel, Analysis and models for small-world graphs, In Proceedings of Symposium on Discrete Algorithms, ACM-SIAM, Vol. 16, 2005.

[14] M. Newman, Models of the small world, J Stat Phys 101 (2000), 819–841.

[15] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, In Proceedings of 32nd ACM Symposium on Theory of Computing, 2000.

[16] D. Chakrabarti, and C. Faloutsos, Graph mining laws, generators, and algorithms, ACM Comput Surv 938 (2006), article 2.

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

[18] R Development Core Team. R: a language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, 2009. http://www.R-project.org

[19] G. Csardi, Network Analysis and Visualization Igraph package for R. Last updated 2012. http://igraph.sourceforge.net/

[20] R. Albert, and A. Barabasi, Statistical mechanics of complex networks, Rev Mod Phys 74, (2002), 47–97.