Published in final edited form as:
J Bioinform Comput Biol. 2012 August ; 10(4): 1250008. doi:10.1142/S0219720012500084.

A Chemical Group Graph Representation for Efficient High-Throughput Analysis of Atomistic Protein Simulations

NOAH C. BENSON[*] and
Division of Biomedical and Health Informatics, University of Washington, Seattle, WA 98195-7240

VALERIE DAGGETT
Department of Bioengineering, Box 355013, University of Washington, Seattle, WA 98195-5013

NOAH C. BENSON: nben@washington.edu; VALERIE DAGGETT: daggett@uw.edu

Abstract

Graphs are rapidly becoming a powerful and ubiquitous tool for the analysis of protein structure and for event detection in dynamical protein systems. Despite their rise in popularity, however, the graph representations employed to date have shared certain features and parameters that have not been thoroughly investigated. Here, we examine and compare variations on the construction of graph nodes and graph edges. We propose a graph representation based on chemical groups of similar atoms within a protein rather than residues or secondary structure and find that even very simple analyses using this representation form a powerful event detection system with significant advantages over residue-based graph representations. We additionally compare graph edges based on probability of contact to graph edges based on contact strength and analyses of the entire graph structure to an alternative and more computationally tractable node-based analysis. We develop the simplest useful technique for analyzing protein simulations based on these comparisons and use it to shed light on the speed with which static protein structures adjust to a solvated environment at room temperature in simulation.

Keywords

graph; network; protein; molecular dynamics; data mining

1. Introduction

Protein structure has been an exciting topic in biology for several years, but as our ability to determine and predict protein structure has improved, it has become clear that protein motions, or dynamics, are an equally important part of the picture. Although several experimental methods exist for studying protein dynamics, molecular dynamics (MD) simulations are the only method of studying protein motions that provide atomic-level resolution at picosecond or finer time resolution in solution. As the cost of computing has decreased over the past several years, MD simulations have become more common, grown longer, tackled larger systems, and become more important in the scientific process. MD simulations are very data rich and, as such, analysis of even a single simulation is complicated, time-consuming, and subjective. The problem is magnified considerably for our Dynameomics Project, which includes multiple native and unfolding simulations of

Correspondence to: VALERIE DAGGETT, daggett@uw.edu.
[*]Current Affiliation: Departments of Neurology and Psychology, University of Pennsylvania, Philadelphia, PA 19104

essentially all autonomous globular proteins in water[22;3] (http://www.dynameomics.org). Such a database can easily eclipse the Protein Data Bank (PDB)[4] in size; for example, the Dynameomics project includes > 10^8 structures—more than 10^4 times as many structures as the PDB. The enormous size of this database, as well as the data density of individual MD simulations, causes the analysis of such datasets to be as difficult a problem as their curation.

To address the problem of efficiently analyzing such large datasets, we explore the use of graphs, or networks, as a means of indexing and searching various properties of MD simulations. We review previous work in this area and compare it to our novel graph-based protein representation. We demonstrate that our representation offers a more detailed treatment of protein dynamics than previous graph methods and show how it can be used for a variety of purposes. Finally, we use these methods to shed light on the time required for a protein in MD simulation to reach a stable conformation in solvent and argue that the commonly used adjustment time of 1 ns is too short.

Graphs are a very natural and efficient tool for representing protein structure due to their ability to capture considerable chemical and steric information in a discrete, compact representation. A graph, **G**, is traditionally defined as an ordered pair of nodes and edges: $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. Conceptually, the nodes tend to represent data points while the edges represent relationships between pairs of nodes ($\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$). In the case of proteins, several graph organizations have been used. Webber *et al.* [25] and Vendruscolo *et al.* [23] represented residues as nodes and drew an edge between two residues whenever their Cα atoms were within a cutoff distance. Amadei *et al.* [1] also represented residues as nodes but drew an edge between a pair of atoms when they had sufficiently correlated motions. Protein graphs have also explored protein topology by representing entire secondary structure elements as nodes and contacts between them as edges[12;10]. Yet another approach has represented entire protein conformations or clusters of conformations as nodes and transitions between these nodes as edges in a Markov state model[6]. For a more complete treatment of protein graph research, readers are encouraged to see reviews by Vishveshwara *et al.* [24], Böde *et al.* [5], and Krishnan *et al.* [13].

Recently, Wriggers *et al.* [27] have chosen a *representative atom* from each amino acid's side chain, based on the work of Singh & Thornton [21], to be represented as a node. The representative atoms of each residue then form the graph's nodes with the graph's edges connecting nodes based either on distance cutoffs or a variant of Delaunay tessellation[9] introduced by Martinetz [16]. By examining the formation and breakage of contacts in the protein's graph over time, they were able to construct an effective activity-monitoring and event-detection metric. Based on their success, we were inspired to explore the use of graphs on a larger scale while adapting the method to retain more of the chemical details of the protein. Another event-detection and online monitoring system was also described recently by Ramanathan *et al.* [17] who used tensor-based principal component analysis (PCA) on sets of all-atom protein graphs. Such event-detection systems are invaluable when dealing with long simulations or large sets of simulations, such as with the Dynameomics database[22;3;19].

Unfortunately, graph representations of proteins have traditionally made several simplifying assumptions, many of which have not been significantly explored. Primary among these has been the choice of nodes in a given graph representation. If one wishes to maximize the chemical information encapsulated by a graph, the natural choice of node is an atom. However, even relatively small proteins can contain more than a thousand atoms, which makes this approach computationally intractable. Accordingly, the most common choice of node in protein representations is the amino acid. This, however, has its own problems. For example, the approach tends to ignore the contributions of backbone atoms, which play very

important roles in protein structure such as forming α-helices and β-strands. Additionally, it ignores the similarities and differences between amino acids pairs such as Leu/Val and Ile/Leu. It is possible to cluster residues into node types, *e.g.* by considering Leu, Val, Ile, and Ala residues to be a single "nonpolar" node type. Yet, even this approach ignores the fact that the Cβ-Cε atoms of Lys can play nonpolar roles in a protein.

Another problem with current protein graph techniques is that they tend to be better at representing static structures than dynamic information; thus they often require a large amount of meta-analysis or considerable transformation to yield useful information. Wriggers *et al.* [27], for example, created a buffer region between contacts to filter out "trivial recrossing" and smoothed the contact distance between each pair of atoms via a moving average. Such approaches yield useful results, but can easily become overly-complicated or computationally intractable. We prefer methods that primarily preserve fidelity as much as possible and that secondarily require as few parameters (*e.g.*, length of buffer region, size of moving average window) as possible. Finally, we prefer our methods to be as computationally efficient as possible. Obviously, it is not possible to eliminate all parameters of an analysis method (*i.e.*, choice of graph representation), but an ideal set of parameters and transformations will produce a set of output data whose relation to the input data (protein simulation) is as direct and simple as possible.

Here, we propose a novel graph representation that partitions a protein into clusters of autonomous chemically-similar atoms. Although this representation is conceptually more complicated than simple amino acid based graph representations, we show that it encapsulates considerably more information and can be used, with extremely simple analysis techniques, to produce valuable metrics and summaries of the dynamical properties of proteins. We test this graph representation on 807 independent protein MD simulations. These 807 proteins are representatives of essentially all (95%) know atonomous protein domains[18].

2. Methods

2.1. Simulations

Although many protocols and programs exist for MD simulations of proteins, we employ our in-house simulation package, *in lucem* Molecular Mechanics (*il*mm)[2] with the Levitt *et al.* [14] force field and explicit three-centered water molecules[15]. Specifically, here we examine the simulations in the Dynameomics database[22], details of which are given elsewhere[3;20;19]. As of April 2010, the Dynameomics database contained >11,000 simulations (> 340 μs) of over 2200 peptides and proteins. Here, we focus on 807 native-state protein simulations (those simulated at 25 °C). These protein folds represent essentially all autonomous globular proteins per our Consensus Domain Dictionary[18;8]. In particular, these simulations represent ~95% of known protein folds, making it an extremely rich and diverse source of simulations for testing novel analysis methods.

2.2. Graphs

Protein structures were saved every picosecond during simulation for analysis, and each structure was transformed into a separate graph. Nodes were built as groups of chemically-similar atoms held in close conformation by covalent bonds (chemical groups). In general, we attempted to keep the size of these groups between 2 and 6 heavy atoms (*i.e.*, small enough that no two atoms in a node would move independently), with some necessary exceptions, such as the Cα of Gly. We defined four node types: nonpolar, dipolar, positive, and negative. Nonpolar nodes consisted of only atoms, primarily carbons, with partial charges, q, such that $|q| < 0.05$. Dipolar nodes consisted of at least two atoms with a weak pole, *e.g.*, the Oδ, Cγ, and Nδ of Asn form a single node. Positive and negative nodes

consisted of atoms with partial charges such that $q > 0.33$ and $q < -0.33$, respectively. Protons were considered to be part of the heavy atom to which they were bonded and contributed to the partial charge of that atom; for example, the Nζ of Lys would normally have a negative partial charge without its attached protons, which serve to make it a positively charged node instead. Atoms that were not covalently bonded were generally not considered to be part of the same node, with the exception of carboxyl groups, whose O atoms were considered a single negative node. A complete visualization of the node definitions, drawn onto a pair of hypothetical peptides with one of each amino acid, is given in Figure 1. Notably, atoms with very high or low partial charges were always separated into positive or negative nodes, even when they had some dipolar character, as the charge of these nodes was considered a more accurate description of them than their polarity. We refer to this graph construction as charge-based graphs in contrast to the traditional residue-based graph construction, in which each residue is a single node.

Edges were drawn between nodes in the protein graphs whenever two atoms in different nodes were in contact; the weight or strength of the edge was the number of pairs of atoms between the two nodes that were in contact, as previously used by Brinda & Vishveshwara [7]. Formally, if nodes u and v contained atoms $u_1, u_2, \ldots u_n$ and $v_1, v_2, \ldots v_m$ respectively, then the strength of edge (u, v), $w((u, v))$, was defined according to Eq. (1), where $\chi(a, b)$ is the contact function, which equals 1 if atoms a and b are within 4.6 Å of each other or if both a and b are carbon atoms and within 5.4 Å of each other; otherwise $\chi(a, b) = 0$.

$$w((u, v)) = \sum_{i=1}^{n} \sum_{j=1}^{m} \chi(u_i, v_j) \quad (1)$$

In order to compare our charge-based graphs to traditional residue-based graphs, we also constructed residue-based graphs using an identical method. The only difference between the residue-based graphs and our charge-based graphs was that in the residue-based graphs there were 20 kinds of nodes, one for each residue type. Each residue, including the protein backbone, was a single node. Two residue-nodes were in contact whenever at least one pair of atoms, one from each residue-node, was in contact. The same source codes were used to generate and analyze both graph types.

2.3. Analysis

Although it is possible and desirable to analyze a raw protein graph, such analyses, which necessarily treat each pairwise contact between nodes as an independent signal, are at best expensive in time and memory and at worst intractable due to the quadratic number of edges in a protein. Further, many simple signal analysis methods, such as principal component analysis (PCA) require quadratic space and time in the number of independent signals resulting in algorithms that are $O(n^4)$ in the number of nodes. Although we examine raw graphs in this paper for comparison, we propose an alternative method of analyzing protein graphs that maintains an $O(n^2)$ complexity.

Two kinds of protein edges, based on the same simulations, were constructed and compared for both charge-based and residue-based graphs. In the first kind of graph, which we refer to as contact-strength graphs or just strength graphs, node-node contacts (edges) were weighted by their strength (Eq. 1). In the second kind of graph, which we refer to as contact-probability graphs or just probability graphs, contact strength was ignored, giving an edge $e = (u, v)$ a weight of $w(e) = 0$ if u and v were not in contact and a weight of $w(e) = 1$ if u and v were in contact (*i.e.*, had a contact strength of at least 1). This weight function was then smoothed using a Gaussian kernel to give a measure of the probability of contact, $w(e, t)$.

The Gaussian kernel had a single parameter, σ, which defined its width; we examined and compared widths of 50 ps, 100 ps, 250 ps, and 500 ps. The smoothing transformed the weight of the edge at time t into the probability that nodes u and v were in contact near time t; more precisely, if a time were randomly chosen from the simulation according to the normal distribution $N(t, \sigma)$, the probability of u and v being in contact at the chosen time would be $w((u, v), t)$.

Both strength-based and probability-based graphs were analyzed using PCA. In order to reduce the complexity of these analyses, the $n(n-1)/2$ signals (one for each pair of nodes) were simplified into n "graph distance" signals (one for each node). For a single node u, the graph distance $d(u, t_0, t_1)$ can be considered a measure of how similar node u's contacts are at time t_0 and time t_1. For the purposes of this measure, all nodes of the same type were considered identical. Although this has the potential to ignore a critical structural change occurring as a node loses contact with one group and gains contact with another similar group, it is both incredibly unlikely that such a change could occur without any significant change in graph distances among all nodes involved and notable that the middle node, in this case, would not have had a drastic change in its environment. Assuming that there are K node types ($K = 4$ for our chemical graphs and $K = 20$ for residue graphs), and that an ordered vector of nodes of type k is given by $N^{(k)}$, then the graph distance is defined by Eq. 2 where $P(N^{(k)})$ is the set of all possible permutations of the ordered list $N^{(k)}$. In practice, this is simpler than it appears; in fact, Eq. 2 reduces to the sum, over all node types, of the Euclidean distance between the *sorted* vectors of edge weights (either strength or probability) between node u and all other nodes of a type type at time t_0 and between u and all other nodes of the same type at time t_1. With efficient data structures, the calculation of graph distances for each node from a reference graph (*i.e.*, the graph of the protein's starting structure) to the graph for each frame of the simulation can be accomplished in $O(mT)$ where m is the number of edges whose weights have non-zero variance over time and T is the number of frames or time-points in the simulation.

$$d(u, t_0, t_1) = \sum_{k=1}^{K} \min_{p \in P(N^{(k)})} \sqrt{\sum_{i=1}^{|p|} \left(w((u, p_i), t_0) - w((u, N_i^{(k)}), t_1)\right)^2} \quad (2)$$

Although graph distance can be used as a generic measurement of how similar a node's contacts are between two different times, or even how similar two completely different nodes' contacts are, we use it here as a metric similar to the root-mean-square deviation (RMSD), in that it tells the extent of change since the beginning of the simulation. By employing graph distance in this fashion, we can perform efficient analyses on the $n \times T$ matrix of graph distances per node over time rather than on the $m \times T$, which is often much larger and can even be intractable.

PCA was performed on all simulations for both strength and probability graphs as well as residue-node graphs. PCA is generally performed on an $m \times n$ matrix where each of the m rows is a dimension and each of the n columns is an observation. For our calculations, each node's graph distance was a dimension and each picosecond was an observation. A projection of each simulation along its principal components (with arbitrary units) was made and examined by changing the base of the graph distance matrix to that of the principal components. Loadings, which are defined as the squared correlation between a signal and a given dimension of the projection of the matrix onto its principal components, were calculated for each node. For example, the first loading for a given node is the squared correlation between the node's graph distance and the first PCA projection over time. The

resulting dataset provides insight into when a protein is undergoing changes in contacts and which nodes are contributing most strongly to those changes.

3. Results

The protein graphs for the 807 Dynameomics targets investigated here ranged in size from 99–1292 nodes, with an average of ~419 ± 217 nodes. The median size was 363.5 nodes. Nodes were 45% nonpolar, 41% dipolar, 5% positively charged, and 9% negatively charged. Most variance in our simulations was due to changes in contacts in the nonpolar and dipolar nodes, while the smallest portion of the variance was in positively charged and negatively charged nodes.

Event detection was remarkably easier using probability-based graphs than strength-based graphs. This was primarily because the first principal components of strength graphs, while usually similar in trend to probability graphs, held considerably more of their variance in high frequency changes rather than low frequency changes. In other words, PCA projections of strength graphs were flatter with more fast fluctuations than probability graphs. Although most proteins in these simulations at 298K exhibit few major changes and thus have relatively flat projections, strength graphs were qualitatively difficult to interpret due to their flatness. Figure 2 shows (a) strength- and (b, c) probability-based PCA projections ($\sigma = 250$ ps) for the proteins Ubiquitin (*1ubq*) and Talin 1 (*1sj7*). In both cases, the strength-based projections are noisy and give no obvious information about the trajectory, while the probability-based projections show clearer trends.

It was not clear that any of the Gaussian kernels used in smoothing the probability graphs ($\sigma = 50$ ps, $\sigma = 100$ ps, $\sigma = 250$ ps, and $\sigma = 500$ ps) were quantitatively superior to any other on the timescale of our simulations (Fig. 2c). There was a slight tendency for principal components of the larger kernels to encapsulate more of the variance than those of smaller kernels, but the difference was small, with the first three principal components of the 500 ps kernel capturing ~ 1% more of the variance than those of the 50 ps kernel. Figure 2c shows PCA projections for all four Gaussian kernels for the proteins Ubiquitin and Talin 1.

The principal component projections of probability-graphs in over two thirds of our simulations revealed an initial period of contact rearrangement. Figure 3 demonstrates this effect with the protein intron endonuclease I-TevI (*1mk0*; Fig. 3a). Figure 3b shows the PCA projections for this simulation, in which almost all of the variance is captured by the first principal component. Notably, the simulation begins with a significant amount of change, culminating around 8 ns, at which point the simulation reaches a more steady state. The loadings of the first principal component projections (Fig. 3c) indicate a patch of residues in and between $\alpha 1$ and $\alpha 2$, which are highlighted in Figure 3a. These residues undergo a large rearrangement starting near 7 ns and ending at 8 ns, with the $\alpha 1$-$\alpha 2$ loop extended and stabilized by the residue Y89. For the remainder of the simulation, this loop, and the rest of the protein, remains stationary and stable in this alternate conformation.

Early contact rearrangement in a simulation followed by relative quiescence, such as shown in Figure 3b, is extremely common, and is expected in any MD simulation due to the process of the protein adjusting to its new thermal and solvent environment and its population of different conformational substates. In fact, this motif occurred in over two thirds of our simulations. Frequently, the first nanosecond (at most) of simulation is ignored in analysis to allow for equilibration. Figure 3 suggests, however, that this may not be long enough in all cases. Notably, this equilibration is typically determined from plots of Cα RMSD over time, whose growth often plateaus within 500 ps. This suggests that our graph analyses reflect more subtle changes on a longer time scale that would be a valuable component in the decision of when analyses should begin.

Not all simulations show the same pattern of initial rearrangement as examined above. Figure 4a shows one such example. This protein, transcriptional regulator ycdc, is a homodimer that is highly stable and shows very little movement during its simulation. The PCA projections (Fig. 4b), however, show small changes throughout the simulation, with the PCA loadings (Fig. 4c) indicating a clump of residues right near residue H36 in the binding pocket (Fig 4a). Closer examination of the protein's entire graph distance matrix (Fig 4d) shows that only the contacts the nodes near H36 change appreciably during the simulation. In fact, H36 flips back and forth between a small cluster of residues and solvent (where the binding pocket is located) during the simulation, creating and breaking contact with its neighbors.

PCA projections of the residue-based graphs usually had a similar form to that of their charge-based counterparts. Several key differences between the two were present, however. Primarily, the residue-based graph analyses were flatter and smoother than their charge-based counterparts and predicted fewer events that the probability graphs. Figure 5a shows a comparison of the three kinds of probability-graph analyses for the β-PIX DBL homology domain (*1by1*). Notably the residue-based PCA projections could almost be a smoothed version of the charge-based PCA projections. The third PCA projection was obtained by performing PCA on the entire matrix of all contact probabilities over time. It follows a very similar trend to that of the graph distance PCA projections with only slight differences. The full probability graph of transcriptional regulator ycdc (Fig 4a) is also shown in Figure 5b along with its residue- and strength-based graph. This full probability graph projection also follows a very similar pattern as that seen in Figure 4b. The strength- and residue-based graph, however, shows almost no resemblance to any other analyses, and was universally the most difficult analysis to interpret.

4. Discussion

Graphs are powerful tools for event detection in MD simulations, as demonstrated here and in previous work[27]. We build on previous work by showing that a more detailed version of the traditional residue-based protein graph better captures the flexible and often duplicitous nature of amino acids. Furthermore, we show that even very simple (and fast) analyses using this particular representation can produce valuable results and provide a graph-based distance metric that can drastically shortcut the computational complexity of more standard graph analyses.

Not surprisingly, strength-based graphs were less effective at detecting events than probability-based graphs. This result is arguably predicted by Wriggers *et al.* [27], who used a similar smoothing mechanism to generate their edges as we have used here. In comparison to probability-graphs, strength-based graphs seemed to carry little usable information. This is despite the fact that strength-based graphs technically encapsulate more information about a simulation than probability-based graphs; *i.e.*, it is trivial to produce a trajectory of probability-graphs from a trajectory of strength-graphs, but it is impossible to perform the reverse because the strength of the contacts is lost. The increased usability of the probability-graphs is likely due to an increased concentration of information via the smoothing process, as supported by the increased variance captured by the initial principal components of analyzed trajectories. Because the edges in probability-graphs convey information about likelihood of contact, they implicitly provide a measure of interactive closeness that is not limited to physically close nodes. Edges in strength-graphs, on the other hand, provide a measure synonymous with physical closeness. In other words, two nodes in a strength-graph are only close if they are physically close, while two edges in a probability-graph are close if they are likely to interact within a given amount of time. The latter

obviously contains more useful dynamic information, even if it is less data-rich than the former.

Notably, strength-graphs have been used effectively for structural analysis[23]. Obviously, probability-graphs cannot be used for the analysis of a single structure by itself, but it is still likely that a single strength-graph would be a more powerful tool for examining a single structure from a simulation than a probability-graph. In this sense, the advantage of the probability-graph over the strength-graph is really its ability to capture information about the *dynamic* interactions of nodes in a protein graph at the cost of information about the precise structure.

Probability-graph analysis, unfortunately, requires choosing an additional parameter, σ. We find that this parameter is rather forgiving, however, and an optimal choice is likely dependent on the speed of the events one wishes to discover with fast events requiring smaller Gaussian kernels. Using a kernel of 250 ps, we were able to catch events that happened on the 0.5–2 ns scale in the binding cleft of a transcription regulator protein (Fig. 4). Additionally, this particular event involved no significant backbone arrangement and no major structural consequence, even among the residues that briefly lost contact with the H36 side-chain. Without the help of an event detection system such as the one employed here, this event would have been virtually impossible to find, especially in a dataset with 807 proteins containing $> 1.7 \times 10^6$ atoms, > 110, 000 residues, and at least 51, 000 structures of each protein.

PCA plots from residue-based graphs were surprisingly flat compared to charge-based graphs when probability edges were used and surprisingly noisy when strength edges were used. Although this may seem initially contradictory, it can be explained by the larger nodes. In the probability graph of a reasonably stable simulation, a residue is unlikely to lose all contact with another residue based on its size. A node of only 2–6 atoms can lose complete contact with another node easily, however. On the flip side, a residue in a strength-based graph can vary the strength of a contact by a large amount even if the two residues remain in contact throughout the simulation. It is clear from the simplicity of probability residue-based graphs and the complexity of strength residue-based graphs that they are both inferior for this kind of analysis to charge-based graphs (Fig. 5). This is not to say that they are useless, however. We hypothesize, that probability residue-based graphs especially may be more useful for extremely large systems, such as those with so many atoms that charge-based analysis becomes intractable or less detail is desired.

The graph distance, especially for probability graphs, was an extremely useful method for reducing computational complexity of our analyses. Without using graph distances, analyses of many of our simulations would have been infeasible. The graph distance metric simplifies a network of contacts into a measure of the change in that contact network as perceived by each individual node. It can be used as an analysis itself (Fig. 4d) and serves as an excellent basis for PCA and other data reduction techniques. It additionally has the advantages of being easy to visually interpret, allowing PCA loadings to point to individual nodes (rather than edges), and automatically incorporating node-type information, which full graph PCA does not. Although a potential drawback of the graph distance is that it can ignore potentially critical rearrangements in a protein if the right contacts form simultaneously, this situation is unlikely. It is, in fact, difficult to imagine a structural change that would be invisible to the graph distance matrix (as opposed to a single node's graph distance over time) yet highly significant for the protein. Analyses of graph distance matrices yielded results that were extremely similar to analyses of the entire graph from which they were derived, demonstrating that it is a robust way to examine protein graphs.

5. Conclusions

Graphs are well-suited to the task of representing protein structural information. We demonstrate the utility of a graph for this task by using a detailed structural representation and employing it, along with very simple analysis techniques, to create an event detection system for protein simulations. We provide evidence that, as far as dynamic information is concerned, graph edges based on the probability of contact between parts of a protein are far more effective than edges based on the strength of the contact and that graphs built with small chemical groups, labeled by their net charge types, are more effective than graphs built with residues. Finally, we show that our graph distance metric is a convenient simplification of the complexity of protein graphs that nonetheless manages to carry most of the information of the original graph. These graph techniques are sufficiently simple to be both scalable and flexible. They can be applied to larger systems and longer simulations and have obvious applications not only to event detection but also to trajectory comparison and docking.

Acknowledgments

We are grateful for support from Microsoft through the External Research Program (to V. D.); the National Library of Medicine through the NIH Training Grant 3 T15 LM007442-04S1 (to N. B.); and the National Institutes of Health, Grant GM 50789 (to V. D.). The MD trajectories contained in the data warehouse were produced using computer time through the DOE Office of Biological Research as provided by the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U. S. Department of Energy under Contract No. DE-AC02-05CH11231. All figures of proteins were produced using Visual Molecular Dynamics (VMD)[11]. All plots were produced using Mathematica 7.0[26].

References

1. Amadei A, Linssen AB, Berendsen HJ. Essential dynamics of proteins. Proteins. 1993; 17(4):412–25. [PubMed: 8108382]
2. Beck, DAC.; Alonso, DOV.; Daggett, V. Technical report. University of Washington; Seattle, WA 98195: 2000–2011. *in lucem* molecular mechanics.
3. Beck DAC, Jonsson AL, Schaeffer RD, Scott KA, Day R, Toofanny RD, Alonso DO, Daggett V. Dynameomics: mass annotation of protein dynamics and unfolding in water by high-throughput atomistic molecular dynamics simulations. Protein: Engineering, Design and Selection. 2008; 21:353–368.
4. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. Nucleic Acids Research. 2000; 28:235–242. [PubMed: 10592235]
5. Böde C I, Kovács A, Szalay MS, Palotai R, Korcsmáros T, Csermely P. Network analysis of protein dynamics. FEBS Letters. 2007; 581(15):2776–82. [PubMed: 17531981]
6. Bowman GR, Huang X, Pande VS. Network models for molecular kinetics and their initial applications to human health. Cell Research. 2010; 20:622–630. [PubMed: 20421891]
7. Brinda KV, Vishveshwara S. A network representation of protein structures: Implications for protein stability. Biophysical Journal. 2005; 89:4159–4170. [PubMed: 16150969]
8. Day R, Beck DAC, Armen RS, Daggett V. A consensus view of fold space: combining scop, cath, and the dali domain dictionary. Protein Science. 2003; 12:2150–2160. [PubMed: 14500873]
9. Delaunay B. Sur la sphère vide. a la memoire de georges voronoi. Izvestiya Akademii Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk. 1934; 7:793–800.
10. Grigoriev IV, Mironov AA, Rakhmaninova AB. Interhelical contacts determining the architecture of alpha-helical globular proteins. Journal of Biomolecular Structure and Dynamics. 1994; 12(3): 559–572. [PubMed: 7727059]
11. Humphrey W, Dalke A, Schulten K. Vmd: visual molecular dynamics. Journal of Molecular Graphics. 1996; 14:33–38. [PubMed: 8744570]
12. Koch I, Kaden F, Selbig J. Analysis of protein sheet topologies by graph theoretical methods. Proteins. 1992; 12(4):314–323. [PubMed: 1579565]

13. Krishnan A, Zbilut JP, Tomita M, Giuliani A. Proteins as networks: usefulness of graph theory in protein science. Current Protein Peptide Science. 2008; 9(1):28–38. [PubMed: 18336321]

14. Levitt M, Hirshberg M, Sharon R, Daggett V. Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution. Computer physics communications. 1995; 91(1–3):215–231.

15. Levitt M, Hirshberg M, Sharon R, Laidig K, Daggett V. Calibration and testing of a water model for simulation of the molecular dynamics of proteins and nucleic acids in solution. J Phys Chem B. 1997; 101(25):5051–5061.

16. Martinetz, T. In: Gielen, S.; Kappen, B., editors. Competitive hebbian learning rule forms perfectly topology preserving maps; Proceedings of the International Conference on Articial Neural Networks (ICANN-93); Heidelberg, Germany: Springer Verlog; 1993. p. 427-434.

17. Ramanathan A, Agarwal PK, Kurnikova M, Langmead CJ. An online approach for mining collective behaviors from molecular dynamics simulations. Journal of Computational Biology. 2010; 17(3):309–324. [PubMed: 20377447]

18. Schaeffer RD, Jonsson AL, Simms AM, Daggett V. Generation of a consensus protein domain dictionary. Bioinformatics. 2011; 27(1):46–54. [PubMed: 21068000]

19. Simms AM, Daggett V. Protein simulation data in the relational model. Journal of Supercomputing. 2011 In Press.

20. Simms AM, Toofanny RD, Kehl C, Benson NC, Daggett V. Dynameomics: design of a computational lab workflow and scientific data repository for protein simulations. Protein: Design, Engineering, and Selection. 2008; 21:369–377.

21. Singh, J.; Thornton, JM. Atlas of Protein Side-Chain Interactions. Vol. I & II. IRL Press; Oxford, United Kingdom: 1992.

22. van der Kamp MW, Schaeffer RD, Jonsson AL, Scouras AD, Simms AM, Toofanny RD, Benson NC, Anderson PC, Merkley ED, Rysavy S, Bromley D, Beck DAC, Daggett V. Dynameomics: a comprehensive database of protein dynamics. Structure. 2010; 18(4):423–35. [PubMed: 20399180]

23. Vendruscolo M, Dokholyan NV, Paci E, Karplus M. Small-world view of the amino acids that play a key role in protein folding. Physical Review E. 2002; 65(6):061,910.

24. Vishveshwara S, Brinda KV, Kannan N. Protein structure: Insights from graph theory. Journal of Theoretical and Computational Chemistry. 2002; 1:187–211.

25. Webber CL, Giuliani A, Zbilut JP, Colosimo A. Elucidating protein secondary structures using alpha-carbon recurrence quantifications. Proteins. 2001; 44(3):292–303. [PubMed: 11455602]

26. Wolfram Research, I. Mathematica. 7.0. Wolfram Research, Inc; Champaign, Illinois: 2008.

27. Wriggers W, Stafford KA, Shan Y, Piana S, Maragakis P, Lindorff-Larsen K, Miller PJ, Gullingsrud J, Rendleman CA, Eastwood MP, Dror RO, Shaw DE. Automated event detection and activity monitoring in long molecular dynamics simulations. Journal of Chemical Theory and Computation. 2009; 5:2595–2605.

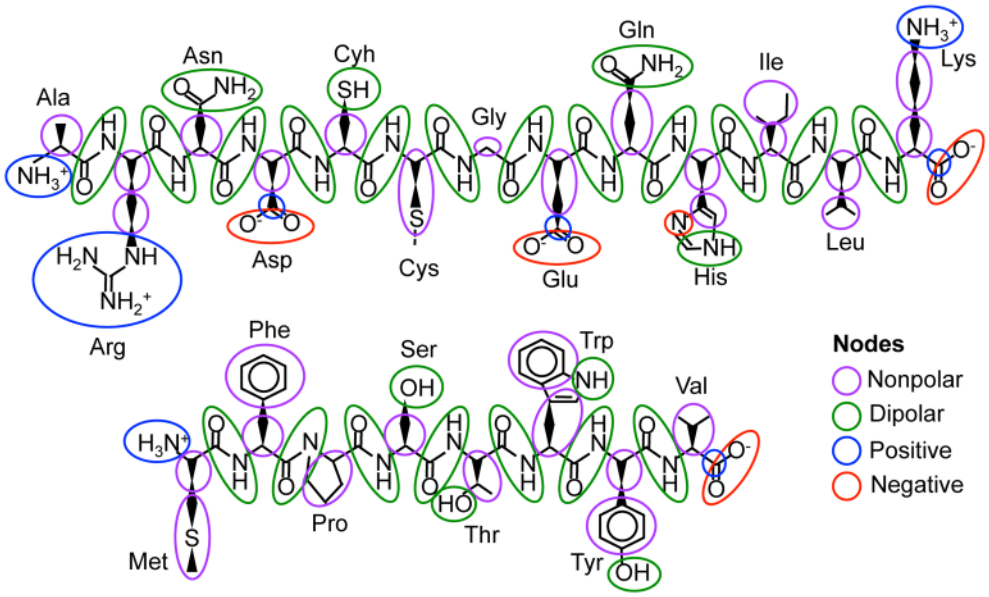

Fig. 1.

Two hypothetical peptide chains, which together include each amino acid and at least one of every possible node for change-based graphs. Nodes are circled by type.

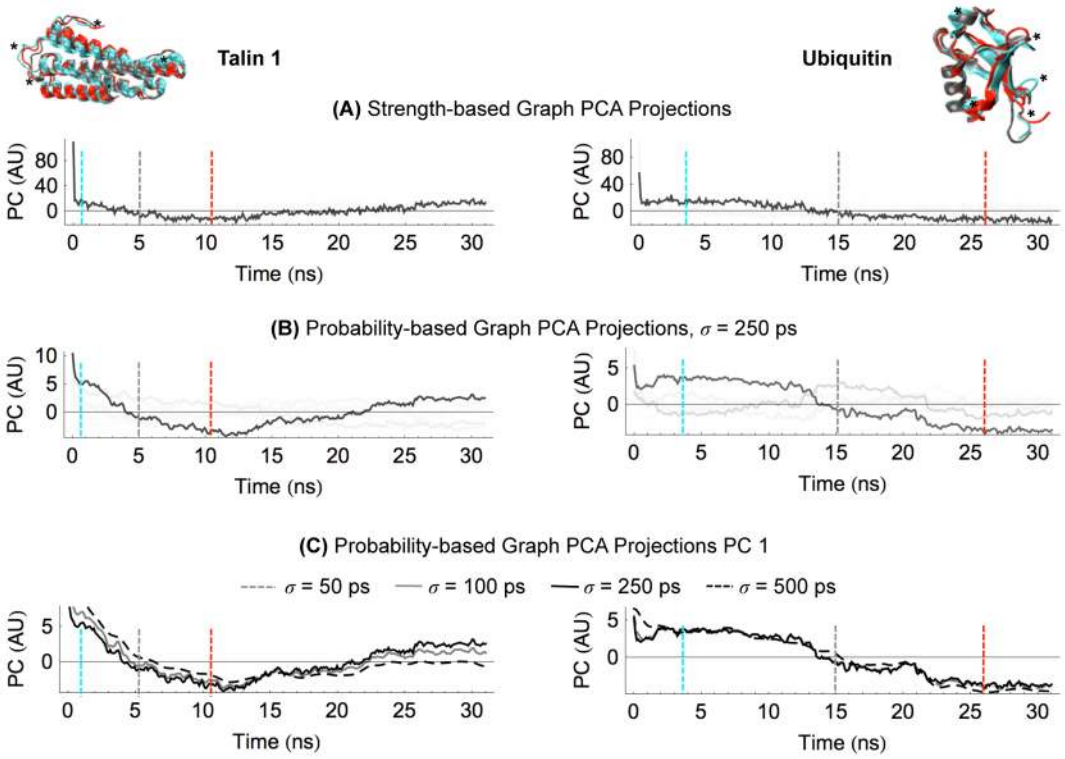

Fig. 2.

Structures and principal component projections of Ubiquitin (*1ubq*) and Talin 1 (*1sj7*). **(a)** Structures and Strength-based graph projections along principal components. Projections are plotted along all dimensions in black, but the opacity of the plot equals the fraction of variance explained by that principal component (throughout). All first component projections encapsulate at least 65% of the variance. The time in the simulation from which structures are taken are indicated by color, and regions of the protein structure with high loading values in the probability-based graph projections are indicated with asterisks. **(b)** Probability-based graph projections with $\sigma = 250$ ps along the three principal components. **(c)** Probability-based graph projections along only the first principal component for four smoothing kernel sizes.

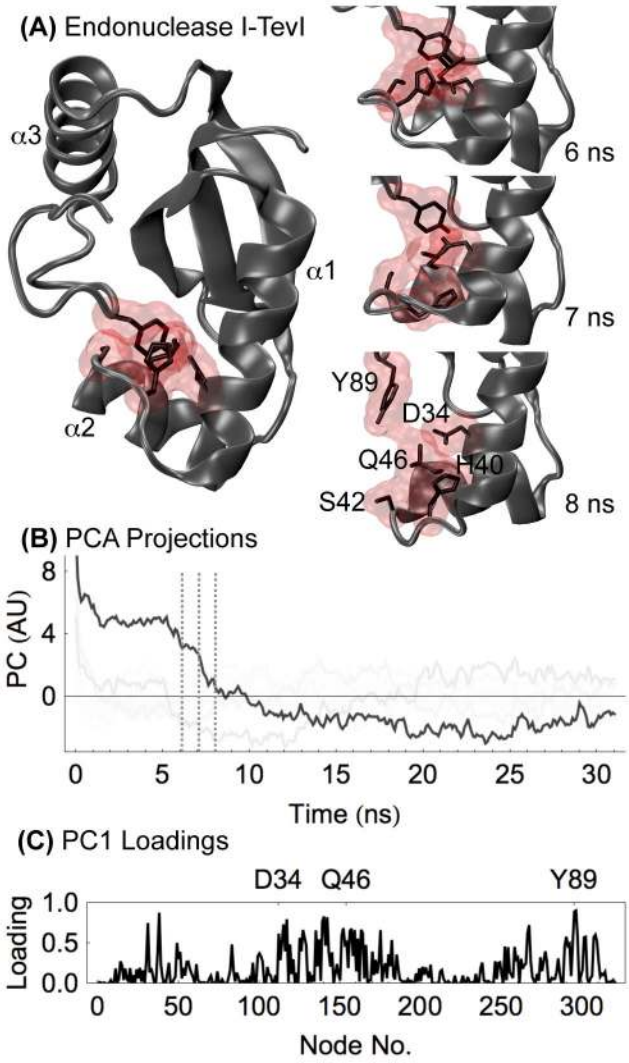

Fig. 3.

The protein intron endonuclease I-TevI (*1mk0*). **(a)** Structures at 0 (left), 6, 7, and 8 ns. Residues D34, H40, S42, L46, S48, and Y89 are shown in both bond and surface forms. **(b)** The projection of the protein's probability-based contacts along the principal components; the graphs for this simulation were made with a Gaussian kernel with parameter $\sigma = 250$ ps. All projected dimensions are plotted in black, but the opacity of each projected dimension is equal to the fraction of variance encapsulated by the that principal component; notable even the second principal component encapsulates < 8% of the variance. **(c)** Loadings of the first principal component projection. Notably, the protein reaches a stable state between 7 and 8 ns in which $\alpha 2$ rotates outward and is stabilized by Y89. After this there is no significant rearrangement in the simulation.

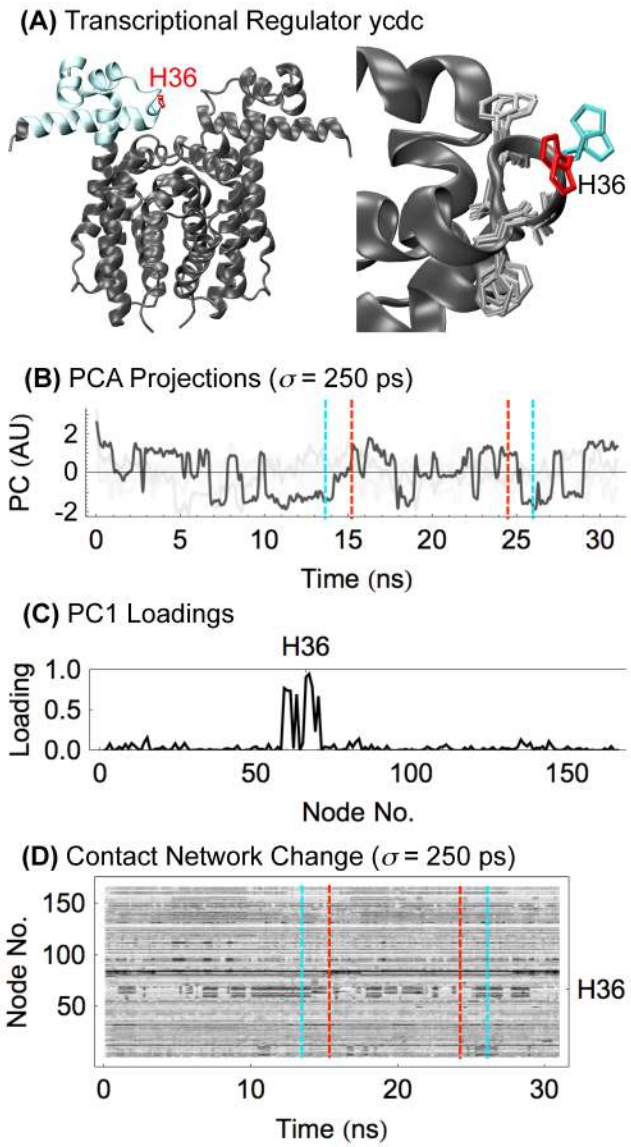

Fig. 4.

The transcriptional regulator ycdc (*1pb6*, superseded by *3loc*). **(a)** Crystal structure of PDB *3loc*, with simulated subunit highlighted in cyan and residue F30 shown in red (left), and structures of the simulated subunit (right). Structures were taken from times 13.5, 15, 24.5, and 26 ns (marked by color in b and d). **(b)** The projection of the protein's probability-based graph distance along its principal components. All principal component projections are plotted in black but with an opacity determined by the proportion of variance captured by that component. **(c)** Loadings of the first principal component. **(d)** The full graph distance matrix for the protein's probability graph using a smoothing kernel of 250 ps. This protein's conformational changes correlate almost perfectly with the movement of residue H36, which switches between the hydrophobic core and a solvent-accessible position in the protein binding site.

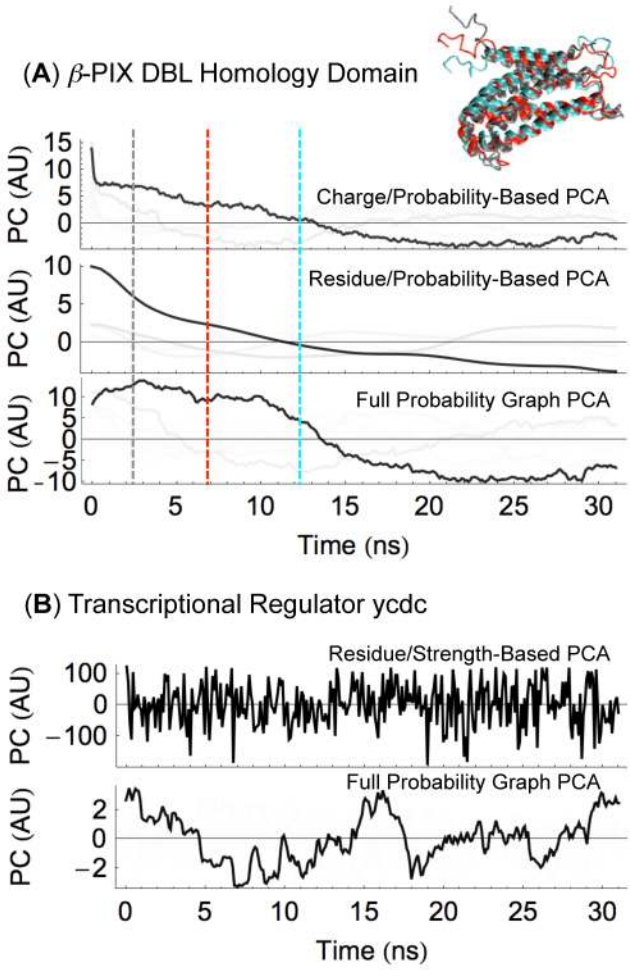

Fig. 5.

Comparison of various graph construction methods. **(a)** Structures and PCA projections of β-PIX DBL homology domain (*1by1*) charge- and probability-based graph (top), residue- and probability-based graphs (middle), and full probability-graph (bottom). **(b)** PCA projections of transcriptional regulator ycdc (*1pb6*; see Fig. 4) strength- and residue-based graph (top) and full probability-graph (bottom). Full probability graphs PCA is performed on the $m \times T$ matrix of all probability edge weights (for all m contact edges) over time (for T ps).