Rigidity Analysis of Protein Molecules

Zahra Shahbazi[1]
Assistant Professor
Department of Mechanical Engineering,
Manhattan College,
Riverdale, NY 10471
e-mail: Zahra.shahbazi@manhattan.edu

Ahmet Demirtas
Department of Mechanical Engineering,
Manhattan College,
Riverdale, NY 10471
e-mail: ademirtas.student@manhattan.edu

Intrinsic flexibility of protein molecules enables them to change their 3D structure and perform their specific task. Therefore, identifying rigid regions and consequently flexible regions of proteins has a significant role in studying protein molecules' function. In this study, we developed a kinematic model of protein molecules considering all covalent and hydrogen bonds in protein structure. Then, we used this model and developed two independent rigidity analysis methods to calculate degrees of freedom (DOF) and identify flexible and rigid regions of the proteins. The first method searches for closed loops inside the protein structure and uses Grübler–Kutzbach (GK) criterion. The second method is based on a modified 3D pebble game. Both methods are implemented in a MATLAB *program and the step by step algorithms for both are discussed. We applied both methods on simple 3D structures to verify the methods. Also, we applied them on several protein molecules. The results show that both methods are calculating the same DOF and rigid and flexible regions. The main difference between two methods is the run time. It's shown that the first method (GK approach) is slower than the second method. The second method takes 0.29 s per amino acid versus 0.83 s for the first method to perform this rigidity analysis.* [DOI: 10.1115/1.4029977]

1 Introduction

Protein molecules are made up of 20 different amino acids. Each amino acid has a basic common structure called main chain or back bone. Main chain includes an amino group at one end and a carboxyl group at the other end which are connected to a carbon atom (α carbon) with an attached hydrogen atom which is shown in Fig. 1. The only difference between various amino acids is a group of atoms connected to α carbon called side chain (R group in Fig. 1). There are 20 different side chain groups, the simplest of which, is a single hydrogen atom [1]. When two amino acids interact, carboxyl group on one amino acid bonds to the amino group of the other one and one water molecule is eliminated. The amino group and carboxyl group of the ends of the protein remain intact. This process continues till all amino acids are connected. The sequence of amino acids is known as protein primary structure [2,3].

After amino acids are all connected, a protein molecule is in its denatured state which resembles a long and open string. When the protein, in its denatured state, is placed in an appropriate environment (i.e., temperature, *p*H, solvent), it folds into a complicated and unique shape called native conformation. The shape change of protein molecule is a result of interatomic forces that occur between the atoms of either the protein or the solvent in which the protein exists [2]. During the folding process, hydrogen bonds are created between nonadjacent amino acids.

By a generally accepted definition, a hydrogen bond is an interaction between a hydrogen atom with an electronegative atom (such as nitrogen and oxygen). It is increasingly recognized that these comparatively weak bonds are connecting main chains and side chains of nonadjacent amino acids and creating rigid domains inside protein structure. Many excellent studies of the hydrogen bonds in proteins [4–8] provided considerable insight into the geometry of hydrogen bonds and their formation conditions. A comprehensive review of the geometry of hydrogen bonds has been conducted in Ref. [9].

Intrinsic flexibility within protein structures allows them to undergo conformational changes [10]. Such changes accommodate different large scale arrangements of protein domains and are essential for protein molecules to perform their tasks while maintaining their structure [11]. Previous efforts have demonstrated that 3D rigid body motion of protein's preserved segments can be assumed as schematic representation for many conformational changes [12–14]. Therefore, identifying rigid and flexible regions of protein molecule is essential to study and consequently replicate and/or control their function to design and fabricate bionanodevices as well as opening new avenues toward new drug design. For instance, identifying the flexible flaps of human immunodeficiency virus (HIV) protease is important for designing new drugs for Acquired Immunodeficiency Syndrome [15].

Computational techniques to analyze and determine flexible and rigid regions of protein molecules are classified in two major categories. The first class of approaches identifies rigid domains by comparing two conformations of a protein [16,17]. This method requires experimental observations of the proteins as well as the atom coordinates, which is an inherently difficult task. In the second class of approaches, rigid domains of a protein are predicted by using a single protein conformation. This category includes molecular dynamics (MD) simulations [18], modal analysis [19,20], graph theory [21–24], and study of amino acids sequence and volume [25,26]. In MD simulations, although protein folding can be simulated with high accuracy, the process is very time consuming [18,27]. Modal analysis uses the eigenvalue analysis. The covalent and hydrogen bonds are replaced by a spring network with a unit spring constant. By calculating zero eigenvalues of the system, independent and redundant bonds can be identified [19,20]. Although, modal analysis is sufficiently accurate, the same problem occurs as does in MD. Karplus and Schulz [25] derived parameters from amino acids sequences and three-dimensional structure of protein molecules, and Ragone et al. [26] combined hydropathy predictions and amino acid volume to predict rigid and flexible regions.

In this study, we represent proteins as nanorigid bodies (different groups of atoms) connected by revolute joints. Then we locate hydrogen bonds using geometric criteria. These bonds are proven to connect these nanorigid bodies and make them rigid with respect to each other. Next, we present two independent graph based approaches to (I) predict DOF and (II) identify rigid regions of protein molecules. Method one originates from GK criterion. In this method first, we found all closed loops inside the 3D structure of mechanism by applying a graph based algorithm. Then using

[1] Corresponding author.

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received November 11, 2014; final manuscript received February 15, 2015; published online April 24, 2015. Editor: Bahram Ravani.

Copyright © 2015 by ASME

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

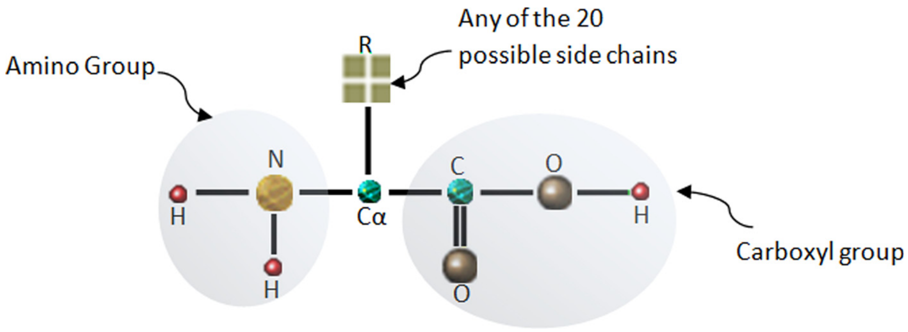


Fig. 1 Amino acid

GK criterion and our developed equations, DOF of each individual loop and consequently the overall DOF of the entire protein is calculated. This method also identifies rigid and flexible regions of the protein. In the second method, we modified three-dimensional pebble game and implemented it on 3D structure of protein molecules and calculated DOF as well as rigid regions of the molecules. To verify these methods, we applied them on several random sample structures. Both methods are producing the same results. Once the methods were verified, we used them to study DOF and rigid regions of protein molecules. The results (DOF and rigid regions) and run time from both methods are compared. It has been shown that the second method takes 0.29 s per amino acid versus 0.83 s for the first method to perform this rigidity analysis.

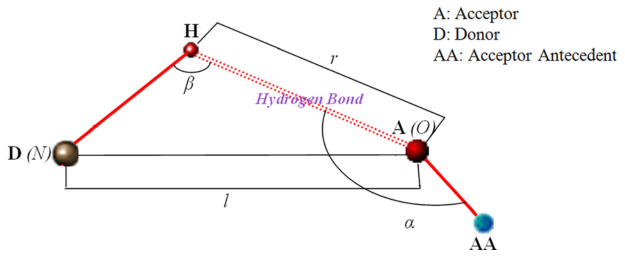


Fig. 2 Geometric parameters of hydrogen bonds

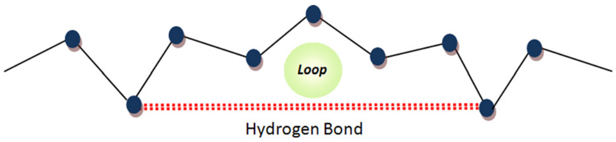


Fig. 3 Closed loop created by a hydrogen bond

2 Rigidity Analysis of Protein Molecules

Both of the two rigidity analysis methods of protein molecules are using ProtoFold, our home developed mechanical model of proteins. ProtoFold models protein molecule as a manipulator and uses direct kinematics (zero reference position method) to provide all atom positions in 3D space [28]. In this model, the back bone of each amino acid has two rigid links and two revolute joints. Each side chain has zero to four rigid links and joints. Then we locate hydrogen bonds inside protein structure based on meeting the suggested geometric criteria in Table 1. Figure 2 shows the geometry of a hydrogen bond. As shown in Fig. 3, a hydrogen bond connects two rigid bodies of nonadjacent amino acids introducing a closed loop into mechanical model. Next, we calculate DOF as well as rigid and flexible regions of protein molecules.

3 Method 1: Rigidity Analysis Using GK Criterion

ProtoFold generates a connectivity matrix including the topological information of the protein molecule. This matrix is parsed to extract all the closed loops. DOF of each of these loops are determined using GK criterion shown in the following equation:

$$\mathrm{DOF} = 6(L-1) - 5J_1 - 4J_2 - 3J_3 - 2J_4 - J_5 \quad (1)$$

where L is the number of links, and J_i is the number of joints with i DOF. Since in protein molecules links are connected only through revolute joints, we just have J_1 and $J_2 = J_3 = J_4 = J_5 = 0$. Then we categorize the closed loops to three groups. (I) loops with less than seven links ($L \leq 6$) which have zero or less DOF. Loops in this group are kinematically over-constrained and can be modeled as independent rigid bodies. (II) Loops with more than six links which are not rigid but constrained. For instance, a loop with eight links and eight joints has 2DOF. (III) Loops that are not rigid and connected to each other and share one or more links. For this category, Eq. (1) can not predict the correct DOF. Therefore, we developed Eq. (2) to determine DOF of these loops

$$\mathrm{DOF} = \sum_{i=1}^{m} (\mathrm{DOF})_i - \sum_{j=1}^{n} P_j \quad (2)$$

where $(\mathrm{DOF})_i$ is the DOF of the ith loop in the group of connected loops and P_j represents jth shared joint. If the DOF of the group of

Table 1 Geometric criteria to detect hydrogen bonds [9]

	α deg	β deg	r(Å)
Main-chain main-chain	[110,180]	[110,180]	<2.5
Main-chain side-chain	[90,180]	[90,180]	<2.5
Side-chain side-chain	[90,180]	[90,180]	<2.5

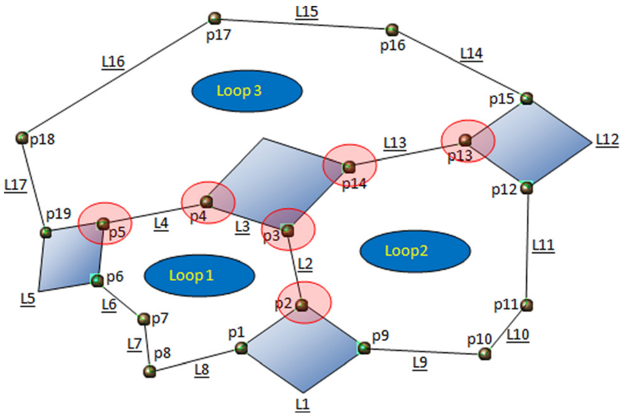
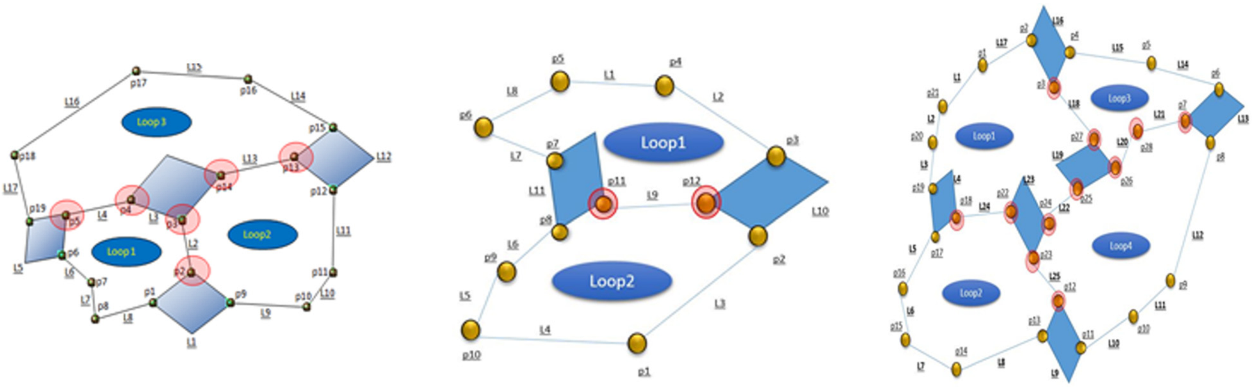


Fig. 4 Three nonrigid loops connected to each other—transparent circles show shared joints

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use



1 DOF

0 DOF

4 DOF

Fig. 5 DOF of different structures calculated using method 1

Table 2 Rigidity analysis on sample protein molecules using methods 1 and 2

	Protein name 1YVQ	2I88	1YVT	2HHB	2K9a	1AUE
No. of amino acids	141	176	141	141	134	91
No. of DOF (no rigidity analysis)	281	354	281	281	267	181
No. of DOF (with rigidity analysis—methods 1 and 2)	43	70	43	61	61	4
Run time method 1 (s)	47.9	107	59.2	52	53.93	18
Run time method 2 (s)	27.12	40	27.4	24	22.16	15.2
% reduction in run time between methods 1 and 2	43	62	53	53	59	16

loops is less than 1, then all of the links of these loops are rigid. In such a case, all links are replaced with one single rigid link. Figure 4 shows an example of connected loops. Here, there are loops 1, 2, and 3 with 2, 2, and 3DOF, respectively, and six common joints. Using Eq. (2), overall DOF will be calculated to be 1.

Figure 5 shows more examples of different structures with connected nonrigid loops. The calculated DOF for these structures using method 1 has been reported in this figure.

To determine overall DOF of protein structure, we first identify rigid loops and replace them with one rigid link. This step is repeated till all the rigid loops are replaced and connectivity matrix is modified. Then we search for nonrigid loops and calculate their DOF. Then each loop with nDOF is replaced with n rigid links and n − 1 joints. Using the new number of links and joints and Eq. (1), DOF of the entire molecule is calculated. Algorithm 1 shows this process. We applied this method on several proteins and the results are shown in Table 2.

4 Method 2: Pebble Game

In this method, we replaced GK criterion with a revised version of "pebble game." Pebble game is a graph theory based algorithm of Laman introduced by Jacobs et al. [21]. We used this method to define rigid regions and DOF of protein structures. In our modified pebble game, the structure is converted into a body-bar (vertex–edge) mechanism. This conversion is built based on kinematic model of protein molecule developed by ProtoFold.

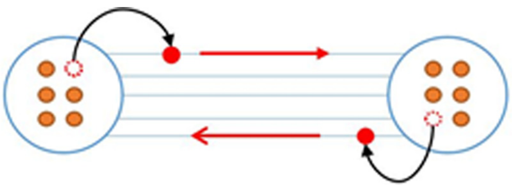


Fig. 6 Pebble movement in specific direction

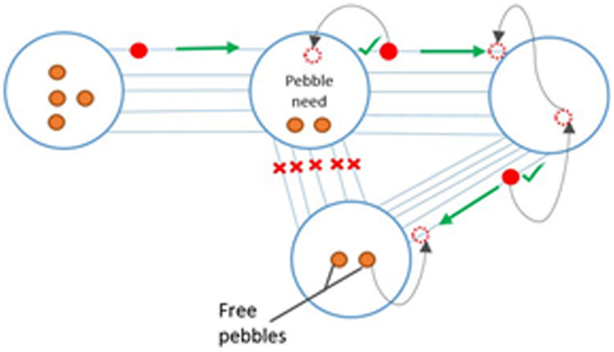


Fig. 7 Path to move a free pebble

Algorithm 1 Pseudocode of the first rigidity method

```
input: Protein structure
output: DOF and rigid regions
Develop mechanical model;
Find hydrogen bonds;
Build connectivity matrix;
Build topological graph;
Find all loops in the graph;
for every loop detected do
        Check loop rigidity according to Grübler–Kutzbach criterion
    if Loop is rigid then
        Replace links of the loop with one rigid link
        Update connectivity matrix
    end
    Identify rigid regions;
    Calculate DOF for nonrigid loops;
    Calculate the new number of links and joints;
    Calculate overall DOF of protein molecule
end
```

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

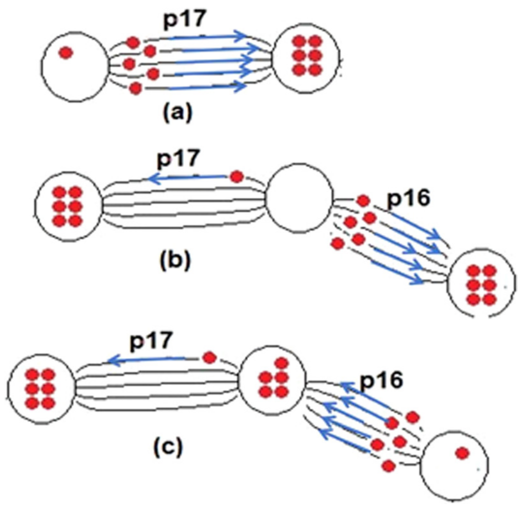


Fig. 8 First pebble motion (*a*), creating a path to neighbor vertex through p16 (*b*), and moving the free pebbles (*c*)

The pebble game can be applied to 2D and 3D structures. In 2D, there are no contradictions and the algorithm can be built upon the equation of $E = 2V - 3$, where E is the number of edges and V is the number of vertices. If this equality holds, it shows the minimally rigid structure. On the other hand, application of 3D structures are a little different than 2D. The original pebble game (FIRST [21]) approach uses atoms and bonds and converts these two into their graph model. This algorithm uses two approaches: (I) both central forces and angular forces or (II) central forces only. For both cases, a vertex has 3DOF and are represented with three pebbles $(E = 3V - 6)$. In our model, we are using Tay theorem $(E = 6V - 6)$.

In our model, first we need to create the graph structure, modeling the 3D mechanism as explained below:

(1) Since each rigid body in 3D space has 6DOF, we assign six pebbles for each vertex. (A rigid body is modeled by a vertex and each pebble represents 1DOF.)
(2) When two vertices are connected, the change in DOF will determine the number of edges between vertices. For instance, when two rigid bodies (each with 6DOF) are connected via a hinge, 5DOF will be removed and the structure will have 7DOF. In this case, five edges will be placed between the vertices which represent the five lost DOF because of the joint between two rigid bodies.

Once the graph structure is created, the pebbles will be moved onto the edges. For two connected vertices, we should have at least seven pebbles in order to move one pebble onto the edge. Each edge can only have one pebble and after a pebble is placed on the edge, in every move this edge will have a pebble placed on it.

When the algorithm is checking an empty edge which is connected to vertices with less than seven pebbles, the algorithm

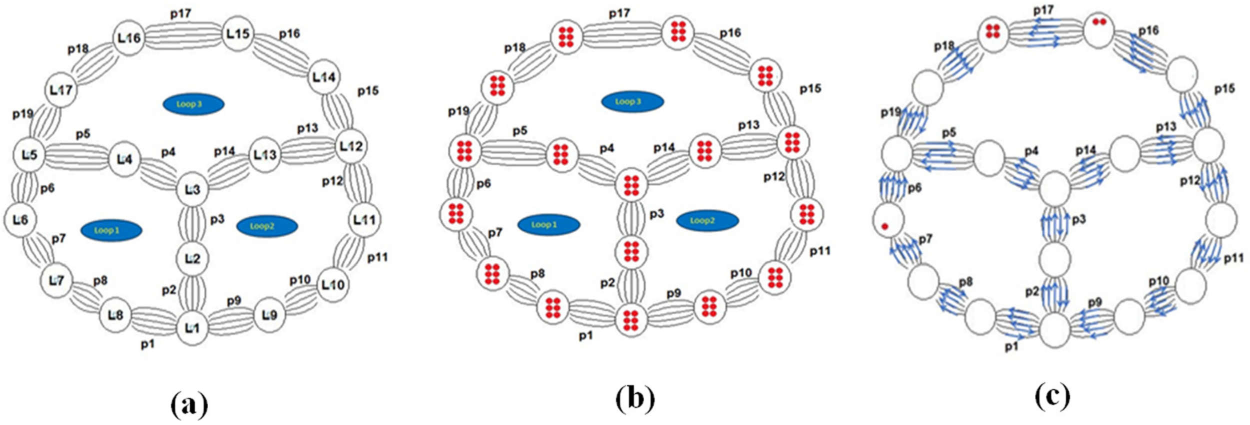


Fig. 9 (*a*) Converted structure for pebble game. (*b*) Pebbles were placed onto each vertex. (*c*) Pebble game result for the given structure.

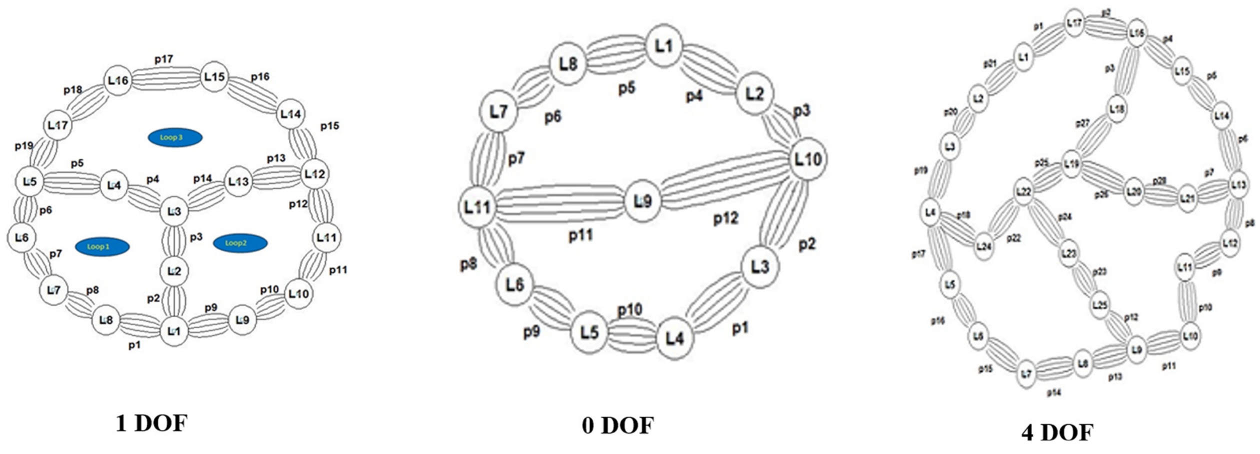
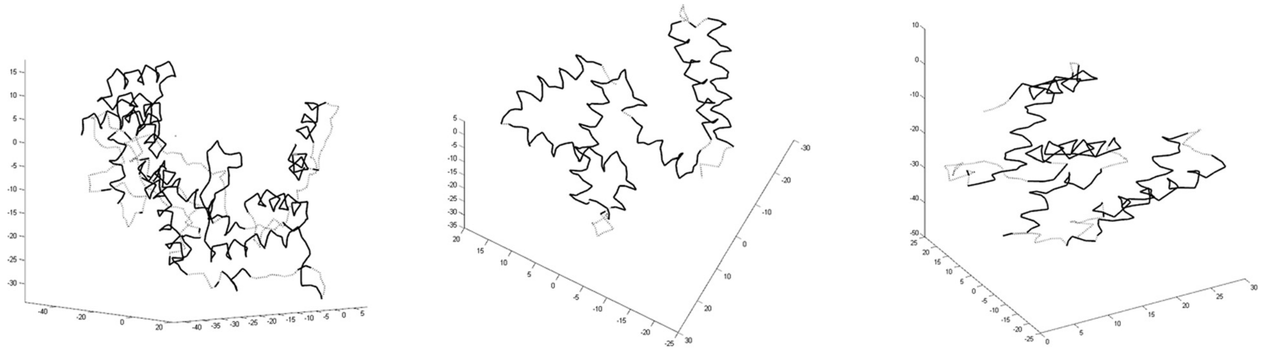


Fig. 10 DOF of various structures using method 2

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use



1JIL 1YVQ 2HHB

Fig. 11 Flexible and rigid regions of sample proteins

searches for free pebbles on neighbor vertices by using the available paths. The pebble will be moved by swapping the paths direction, if the algorithm finds free pebbles. If we cannot reach any free pebble the edge will be called overconstrained (redundant) region. This movement as shown in Fig. 6 should happen in specific direction. Figure 7 shows the specific path used to move one free pebble. For moving other pebbles, other paths are to be found.

We can define the DOF by counting the free pebbles, after all the edges are observed. If the total number of free pebbles is more than six, the extra pebbles will define the DOF. For instance, the structure will have 1DOF if the total of free pebbles is 7. On the other hand, if total number of free pebbles is six we can define the structure as a rigid body (0DOF). When we have edges that do not have pebble on them, they are overconstrained regions. The pseudocode of the procedure is provided in algorithm 2.

Here, we explain the algorithm using the example shown in Fig. 4. First, we modify the structure by converting joints to edges and links to vertices (Fig. 9(*a*)). Then six pebbles are placed on each vertex (Fig. 9(*b*)) and pebble movement starts. As seen in Fig. 8(*a*), pebbles move to the edges. The process continues until all the edges checked. If the algorithm faces a situation as shown in Fig. 8(*b*) where the structure has more then six pebbles and at the same time there is at least one empty edge which is not identified as redundant edge the algorithm can locate the free pebbles on next neighbor vertex. As mentioned previously, the algorithm creates a path to the neighbor(s) to get the free pebbles. The arrows represent the path direction to neighbors. Since all of the arrows on p16 allow us to take the free pebbles, these pebbles can be moved onto p16 and previously placed pebbles can be taken back as seen in Fig. 8(*c*) by using same pebble movement process. Also, when the free pebbles are moved, the arrow directions are changed (edge p16 Figs. 8(*a*) and 8(*b*)).

Algorithm 2 Pseudocode of the modified pebble game. Here, *V* is the number of vertices, *E* is the number of empty edges, and *C* is the number of free pebbles, and *u*, *v* are the vertices connected to each other with same edge

```
      input: Connectivity matrix (CM)
      output: Rigid sub graphs and DOF
1     Use first row or column to define vertices;
2     Place 6 Free pebbles on each vertex $C = 6 \times V$;
3     From CM create 5 empty edge connections between vertices;
4     Define all edge as a matrix “Empty Edge (EE);”
5     while $EE > 0$ do
6         if vertices u(*) and v(*) have more than 6 pebbles then
7                 Place 1 free pebble on an empty edge from any of this vertices by directing the edge out from the vertex ($C = C - 1$)
8                 Change the identification of edge as independent
9                 Move to next edge
10                Go to 6
11        else
12            Search for free pebbles by following the directed edges from u or v
13            if free pebble x found (except the pebbles on u and v) then
14                By using the path (P), move the x with swapping the direction of P until this free pebble appears on the u or v
15            else
16                Change the identification of the edge as redundant
17                if there are free pebbles and a path is available to free pebble then
18                    Move to next edge
19                    Go to 6
20                else
21                    if there are free pebbles and a path is not available to the free pebbles then
22                        Define the region as rigid
23                        Move to next edge
24                        Go to 6
25                    end
26                end
27            end
28        end
29    end
30    To define the DOF count the free pebbles DOF = $C - 6$.
```

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

When the algorithm continues to check every single edge, the structure will end up with total of seven pebbles as shown in Fig. 9(*c*). By having this information, DOF can be found to be one. We applied this method on the same sample structures shown in Fig. 5. The results are the same as method one and are shown in Fig. 10.

We then applied this method on the same sample protein molecules which we used for method 1. The DOF calculated by method 2 (pebble game) are shown in Fig. 2. Also, the rigid and flexible regions of some sample proteins are presented in Fig. 11. Here, the dashed line shows the flexible regions and the solid line identifies rigid regions.

5 Discussion and Conclusion

Stability of protein molecules is achieved by rigid regions inside of their structures. At the same time, flexible regions let the protein molecules move, change their conformations and perform their biological tasks. Therefore, predicting rigid and flexible regions of protein molecules is inevitable to understand the mobility and consequently their functions. At the same time, taking advantage of protein mobility analysis results in a more realistic as well as computationally faster simulation of protein molecules as kinematic chains.

One of the critical factors in protein mobility analysis is the formation of hydrogen bonds. In this work, we identify hydrogen bonds based on geometric criteria then study how these bonds create rigid and flexible closed loops. We developed two methods to detect flexible and rigid regions of protein molecules as well as their DOF. We employed both methods on simple mechanical structures as well as complicated protein structures and achieved the same results from both methods. This analysis shows that DOF of protein molecules has been dramatically changed considering hydrogen bonds in mechanical model. This, in turn, makes the mechanical model more reliable because of considering hydrogen bonds. Also, it can provide immediate computational benefit to protein folding analysis by dramatically reducing DOF of protein kinematic model.

The significant difference between the two methods is the speed of the calculations. Method 1 takes an average of 0.83 s per amino acid versus 0.29 s for the first method. Considering the fact that the protein molecules which undergo conformational changes are usually large molecules, method 2 has an obvious advantage over the first method.

References

[1] Gutteridge, A., and Thornton, J., 2005, "Understanding Nature's Catalytic Toolkit," Trends Biochem. Sci., **30**(11), pp. 622–629.

[2] Branden, C., and Tooze, J., 1999, *Introduction to Protein Structure*, 2nd ed., Garland Publishing, New York.

[3] Murray, R., Harper, H., Granner, D., Mayes, P., and Rodwell, V., 2006, *Harper's Illustrated Biochemistry*, McGraw-Hill, Columbus, OH.

[4] Baker, E. N., and Hubbard, R. E., 1984, "Hydrogen Bonding in Globular Proteins," Prog. Biophys. Mol. Biol., **44**(2), pp. 97–179.

[5] Mcdonald, I. K., and Thornton, J. M., 1994, "Satisfying Hydrogen-Bonding Potential in Proteins," J. Mol. Biol., **238**(5), pp. 777–793.

[6] Kortemme, T., Morozov, A. V., and Baker, D., 2003, "An Orientation-Dependent Hydrogen Bonding Potential Improves Prediction of Specificity and Structure for Proteins and Protein-Protein Complexes," J. Mol. Biol., **326**(4), pp. 1239–1259.

[7] Xu, D., Tsai, C. J., and Nussinov, R., 1997, "Hydrogen Bonds and Salt Bridges Across Protein-Protein Interfaces," Protein Eng., **10**(9), pp. 999–1012.

[8] Alexandrescu, A. T., Snyder, D. R., and Abildgaard, F., 2001, "NMR of Hydrogen Bonding in Cold-Shock Protein A and an Analysis of the Influence of Crystallographic Resolution on Comparisons of Hydrogen Bond Lengths," Protein Sci., **10**(9), pp. 1856–1868.

[9] Shahbazi, Z., Ilies, H., and Kazerounian, K., 2010, "Hydrogen Bonds and Kinematic Mobility of Protein Molecules," ASME J. Mech. Rob., **2**(2), p. 021009.

[10] Van Holde, K., and Mathews, C., 1996, *Biochemistry*, Benjamin-Cummings Pub. Co., Menlo Park, CA.

[11] Lodish, H., Berk, A., Matsudaira, P., Kaiser, C., Krieger, M., Scott, M., Zipurksy, S., and Darnell, J., 2004, *Molecular Cell Biology*, 5th ed., WH Freeman and Company, New York.

[12] Hinsen, K., 2008, "Structural Flexibility in Proteins: Impact of the Crystal Environment," Bioinformatics, **24**(4), pp. 521–528.

[13] Kumar, S., Wolfson, H. J., and Nussinov, R., 2001, "Protein Flexibility and Electrostatic Interactions," IBM J. Res. Dev., **45**(3–4), pp. 499–512.

[14] Najmanovich, R., Kuttner, J., Sobolev, V., and Edelman, M., 2000, "Side-Chain Flexibility in Proteins Upon Ligand Binding," Proteins-Struct. Funct. Genet., **39**(3), pp. 261–268.

[15] Heal, J. W., Jimenez-Roldan, J. E., Wells, S. A., Freedman, R. B., and Römer, R. A., 2012, "Inhibition of HIV-1 Protease: The Rigidity Perspective," Bioinformatics, **28**(3), pp. 350–357.

[16] Nichols, W. L., Rose, G. D., Ten Eyck, L. F., and Zimm, B. H., 1995, "Rigid Domains in Proteins: An Algorithmic Approach to Their Identification," Proteins, **23**(1), pp. 38–48.

[17] Wirggers, W., and Schulten, K., 1997, "Protein Domain Movements: Detection of Rigid Domains and Visualization of Hinges in Comparisons of Atomic Coordinates," Proteins, **29**(1), pp. 1–14.

[18] Sinha, N., and Smith-Gill, S., 2005, "Molecular Dynamics Simulation of a High-Affinity Antibody-Protein Complex," Cell Biochem. Biophys., **43**(2), pp. 253–273.

[19] Suhre, K., and Sanejouand, Y. H., 2004, "ElNemo: A Normal Mode Web Server for Protein Movement Analysis and the Generation of Templates for Molecular Replacement," Nucleic Acids Res., **32**(Suppl. 2), pp. W610–W614.

[20] Doruker, P., Bahara, I., Baysalb, C., and Erman, B., 2002, "Collective Deformations in Proteins Determined by a Mode Analysis Molecular Dynamics Trajectories," Polymer, **43**(2), pp. 431–439.

[21] Jacobs, D. J., Rader, A. J., Kuhn, L. A., and Thorpe, M. F., 2001, "Protein Flexibility Predictions Using Graph Theory," Proteins, **44**(2), pp. 150–165.

[22] Jacobs, D., and Thorpe, M., 1995, "Generic Rigidity Percolation: The Pebble Game," Phys. Rev. Lett., **75**(22), pp. 4051–4054.

[23] Lee, A., Streinu, I., and Theran, L., 2008, "Analyzing Rigidity With Pebble Games," Proceedings of the 24th Annual Symposium on Computational Geometry, pp. 226–227.

[24] Lee, A., and Streinu, I., 2005, "Pebble Game Algorithms and $(k, l)-$ Sparse Graphs," 2005 European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05), Stefan Felsner, ed., Discrete Mathematics and Theoretical Computer Science Proceedings AE, pp. 181–186.

[25] Karplus, P. A., and Schulz, G. E., 1985, "Prediction of Chain Flexibility in Proteins," Naturwissenschaften, **72**(4), pp. 212–213.

[26] Ragone, R., Facchiano, F., Facchiano, A., Facchiano, A. M., and Colonna, G., 1989, "Flexibility Plot of Proteins," Protein Eng., **2**(7), pp. 497–504.

[27] Poursina, M., 2014, "Modified Fast Multipole Method for Coarse-Grained Molecular Simulations," Biophys. J., **106**(2), p. 407a.

[28] Kazerounian, K., Latif, K., and Alvarado, C., 2005, "Protofold: A Successive Kinetostatic Compliance Method for Protein Conformation Prediction," ASME J. Mech. Des., **127**(4), pp. 712–717.

Downloaded From: https://computingengineering.asmedigitalcollection.asme.org on 07/01/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use