PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Method for identification of rigid domains and hinge residues in proteins based on exhaustive enumeration

Jaehyun Sim,[1†] Jun Sim,[2†] Eunsung Park,[3] and Julian Lee[2*]

[1] Department of Oral Microbiology and Immunology, School of Dentistry, Seoul National University, Seoul 110-749, Korea

[2] Department of Bioinformatics and Life Science, Soongsil University, Seoul 156-743, Korea

[3] Administrative Service Division, Apsun Dental Hospital, Seoul 135-590, Korea

ABSTRACT

Many proteins undergo large-scale motions where relatively rigid domains move against each other. The identification of rigid domains, as well as the hinge residues important for their relative movements, is important for various applications including flexible docking simulations. In this work, we develop a method for protein rigid domain identification based on an exhaustive enumeration of maximal rigid domains, the rigid domains not fully contained within other domains. The computation is performed by mapping the problem to that of finding maximal cliques in a graph. A minimal set of rigid domains are then selected, which cover most of the protein with minimal overlap. In contrast to the results of existing methods that partition a protein into non-overlapping domains using approximate algorithms, the rigid domains obtained from exact enumeration naturally contain overlapping regions, which correspond to the hinges of the inter-domain bending motion. The performance of the algorithm is demonstrated on several proteins.

Proteins 2015; 83:1054–1067.

Key words: protein structure; rigid domain; hinge; graph theory; maximal clique; exact; exhaustive enumeration; normal mode analysis; elastic network model; DAGR.

INTRODUCTION

Protein structures are dynamic and often undergo conformational changes in order to perform their functions. For example, there are many proteins that change conformations upon binding with ligands. When performing a docking simulation for such a protein, it is crucial to take the flexibility of the receptor into account.[1–8] There are many proteins whose large-scale motions are described by relative movements of relatively rigid substructures, called the rigid domains. It is important to identify the rigid domains, and the hinge residues important for the inter-domain motion, in order to understand the large-scale motion of such a protein. A natural definition of a rigid domain is a subset of protein residues in which the distance between any pair of residues within this subset is fixed.[9,10] This definition is also consistent with the standard definition of a rigid body in physics, and forms the basis for an alternative definition of protein domains.[11] When multiple conformations for a given protein are deposited in the Protein Data Bank (PDB),[12] then it is in principle straightforward to detect rigid domains by comparing these distinct conformations.

However, although many methods have been developed that compute rigid domains from multiple conformations,[9,10,13–17] computations usually have been based on some approximations. One exception is Ref. [9], where the authors computed all the rigid substructures of human hemoglobin, a protein with a chain length of 42 amino acids, by comparing its oxy (1HHO) and deoxy (2HHB) forms. However, in the end, the authors concluded that such an exhaustive enumeration involves a prohibitively high computational cost for use

Additional Supporting Information may be found in the online version of this article.

Grant sponsor: Korea Healthcare Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health and Welfare, Republic of Korea; Grant number: HI12C0675.

[†]Jun Sim and Jaehyun Sim contributed equally to this work.

*Correspondence to: Julian Lee, Department of Bioinformatics and Life Science, Soongsil University, Seoul 156-743, Korea. E-mail: jul@ssu.ac.kr

Received 3 October 2014; Revised 28 February 2015; Accepted 10 March 2015
Published online 27 March 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/prot.24799

as a general method and developed a rather ad-hoc heuristic method for identifying rigid domains. On the other hand, it is clear that any subset of a rigid domain is also a rigid domain, and therefore rigid domains fully contained within others are rather trivial and provide no useful information. Therefore, by restricting the analysis only to maximal rigid domains that are not proper subsets of other rigid domains, one can significantly reduce computational cost.

In this work, we develop a method for rigid domain detection called DAGR (Domain Analysis based on GRaph theory), based on the exhaustive enumeration of all the maximal rigid domains. The task is performed by mapping the maximal rigid domains to maximal cliques in graph theory, where we can utilize efficient and exact algorithms for identifying all the maximal cliques.[18,19] A set of rigid domains are then selected among the maximal rigid domains, which cover most of the protein with minimal overlap. The common residues of these domains are the hinge for the relative bending motion of the domains. This is in contrast to most of the previous methods[10,13–17] that partition a protein into nonoverlapping rigid domains by deliberately removing information on overlapping regions.

The idea of graph theory has been used previously for analyzing protein motion,[20] flexibility,[21–28] and allostery.[29] The clique concept in graph theory has also been applied to various topics in bioinformatics and chemoinformatics, such as gene clustering,[30,31] ecological modeling,[32] phylogeny,[33] protein structure prediction,[34] protein structure alignment,[35] protein functions and interactions,[36,37] and the classification of chemicals.[38] However, the clique concept has not been utilized thus far for the computation of rigid domains in proteins.

In the following sections, we first elaborate on the method, and then demonstrate the performance of our method by applying it to several proteins in which rigid domains and hinge residues are successfully identified.

METHODS

Exhaustive enumeration of maximal rigid domains using graph theory

As explained earlier, a rigid domain is a subset of protein residues in which the distance between each pair of residues is invariant when two conformations are compared. Let us consider a protein of size N_s, and define the distance between a pair of residues as the distance between their C_α atoms. Because the inter-residue distances are real-valued quantities with limited precision, no two distances can be exactly equal. Therefore, two distances are considered as equal when the absolute value of their difference is less than a predefined cutoff d_{cut}. Therefore, given two conformations A and B of a protein with C_α coordinates $\{\mathbf{r}_i^A\}$ and $\{\mathbf{r}_i^B\}(i=1,\ldots,N_s)$, a rigid domain is defined as a subset $S=\{i_1,\cdots i_k\}(k \le N_s)$ where[9,10]

$$||\mathbf{r}_i^A-\mathbf{r}_j^A|-|\mathbf{r}_i^B-\mathbf{r}_j^B|| < d_{\mathrm{cut}} \qquad (1)$$

for all pairs $i,j \in S$.

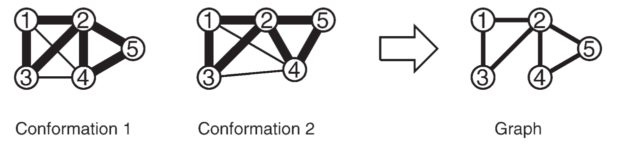

Figure 1

Example of two conformations and the resulting graph. Invariant inter-residue distances are denoted by thick lines and variant ones by thin lines. In the graph, the cliques are {1},{2},{3},{4},{5},{1, 2},{1, 3},{2, 3},{2, 4},{2, 5}, {4, 5},{1, 2, 3}, and {2, 4, 5}, but only {1, 2, 3} and {2, 4, 5} are the maximal cliques, which also happen to be the maximum cliques. Residue 2 is common to the two maximal cliques and is the hinge residue.

Some methods identify rigid domains of protein structures using the minimum root mean square deviation (RMSD) values of the coordinates.[13–15,17] Although such a method will give the same results if the protein consists of ideal rigid domains, it has been shown[9,10] that in the case of real proteins, the usage of the definition of the rigid domains based on inter-residue distances, Eq. (1), leads to more sensitive methods than those based on RMSD fit.

We now map the protein into a graph with N_s vertices, where each vertex represents a residue, and an edge exists between a pair of vertices if and only if the condition (1) holds for the pair. Then, a rigid domain corresponds to a subgraph in which all the vertices are connected with the other vertices. This is called a clique in graph theory. As mentioned earlier, we are interested in rigid domains that are not fully contained in others, so we compute cliques that are not proper subsets of other cliques. These are called maximal cliques.[1] The terminology is summarized in Figure 1.

The algorithm for enumerating the maximal cliques of a given graph has been developed by Bron and Kerbosch.[18,19] Combined with the efficient coding in C language where the set operations in the Bron–Kerbosch algorithm is implemented as bit-wise operators, the maximal rigid domains up to order of 10^6 can be easily generated within several seconds, as will be shown in the results section.

[1] Maximal cliques are more general than maximum cliques, which are cliques with the maximum size. Maximum cliques and maximal cliques correspond to global and local maxima. Because the set of maximum cliques is a subset of the set of maximal cliques, the exhaustive enumeration of maximal cliques is a much more difficult problem than that of finding maximum cliques.

Selection of maximal rigid domains with maximal coverage and minimal overlap

The set of all the maximal rigid domains cannot be used directly, since there are usually too many of them with much redundancy, containing many similar maximal rigid domains with huge overlap that differ only by a few residues. Therefore, it is of interest to remove this redundancy and find a set of maximal rigid domains D_i $(i = 1, \cdots, n)$ that cover most of the protein with minimal overlap. That is, for a given value of n, D_is are selected so that the size of their union

$$C \equiv \cup_{i=1}^{n} D_i \qquad (2)$$

is maximized. Among the domain set $\{D_i\}$ with the same maximal size of C, the set with minimal overlap is chosen, where the overlapping residues are defined as the union of common residues between each pair of the domains:

$$H \equiv \cup_{i<j}(D_i \cap D_j). \qquad (3)$$

One of the difficulties of this procedure is that one usually does not know beforehand how many maximal rigid domains can result in satisfactory coverage of the protein structure, so n has to be found by trial and error. Furthermore, the computation time scales as $O(N^n)$ where N is the total number of maximal rigid domains, so the procedure is computationally rather costly for large values of n. Therefore, we constructed an approximate procedure in which the candidate domains are iteratively selected one at a time, with previously selected domains remaining fixed. Let us denote the union of all the maximal rigid domains selected after the kth iteration as C, and their overlapping regions as H, defined by Eqs. (2) and (3) where n is replaced by k. At the next iteration, the maximal rigid domain D with the largest size of $C \cup D$ is chosen, and among those with the same maximal size of $C \cup D$, the one with the smallest size of $H \cup (C \cap D)$ is selected. Because $C = \varnothing$ at the beginning, the first domain selected is the one with the largest size. The iteration continues until satisfactory coverage is achieved. This iterative procedure is similar to the algorithm employed in RigidFinder[10] where the largest rigid domain is searched and removed from the sequence iteratively. The main difference is that, in the event that more than one domain can be selected according to this criterion, the one with minimal size of the overlapping region is chosen in the current method, whereas the choice is rather arbitrary in the case of RigidFinder. Additionally, note that the exact maximal rigid domains are obtained here, in contrast to RigidFinder where an approximate algorithm is employed.

The definition of the hinge as the overlapping region of rigid domains is not only natural but also provides useful information on the nature of inter-domain motion. For example, the absence of overlapping residues at a domain boundary implies that the motion cannot be described as a bending relative to this boundary, as in the case of a relative translational motion.

Postprocessing

Most of the previous methods for rigid domain detection[10,15] perform some kind of coarse-graining so that the reported rigid domains are smooth, in contrast to the raw results of DAGR that contains short fragments and gaps (see Results). Therefore, for a close comparison with other methods, we also optionally performed postprocessing according to the prescription of Rigid Finder[10]: We filled gaps and removed fragments of lengths less than four, where those with shorter lengths were treated before those with longer lengths, and gaps with a given length were filled before the fragments with the same lengths were removed. When performing the postprocessing, any sets of residues located >10 Å from each other in physical space are considered as separate domains.[10]

Prediction of rigid domains via the generation of an alternative conformation

Although multiple conformations are needed to compute rigid domains, rigid domains also can be predicted from a single conformation.[16,17,22,25,27,39–46] DAGR can be adapted for this purpose easily once fluctuations in the protein structure have been predicted, using methods such as normal mode analysis[16,17,39–41,43–45,47–83] or molecular dynamics simulations.[46,84] Once a large-scale motion has been predicted, it is then straightforward to draw a graph of distance constraints and enumerate maximal cliques. Here, we employed a simple method of normal mode analysis, the elastic network model (ENM).[52–83] Some earlier methods for rigid domain predictions used a simpler version of ENM, the Gaussian network model (GNM); in the GNM, only information about the inter-residue coupling strengths is used.[41,53,54] To use the graph theory formalism that we developed, the predicted alternative conformation should be explicitly constructed.[77] Therefore, we used an anisotropic network model (ANM),[52,55] from which directions of motion for individual residues can be predicted. In this model, springs of equal strength are attached to each nonadjacent C_α pair, whose distances are closer than a given cutoff value. The cutoff value was set to 10 Å, which is similar to the cutoff values that have been used previously.[41,45,77,78] Note that this cutoff includes adjacent C_α carbons, whose distances are almost fixed due to the rigidity of the covalent bonds. To implement this approximate invariance in length, we set the strength of the spring between adjacent C_α pairs as 100 times that of the other springs. A preliminary test suggested that the result does not depend significantly on the precise value of this ratio (data not shown). Given that the equilibrium length of the spring is taken as the distance provided in the input structure, the input structure is the minimum of the potential energy by construction. Denoting the original position and the fluctuation of the ith C_α atom as $\mathbf{R}_i$ and $\mathbf{r}_i$, respectively, the pairwise potential energy between the ith and jth position is

$$\begin{aligned} E_{ij}(\mathbf{r}_i,\mathbf{r}_j) &= \frac{k}{2}(|\mathbf{R}_i+\mathbf{r}_i-\mathbf{R}_j-\mathbf{r}_j|-|\mathbf{R}_i-\mathbf{R}_j|)^2 \\ &= \frac{k}{2}\left(\sqrt{|\mathbf{R}_{ij}|^2+|\mathbf{r}_{ij}|^2+2\mathbf{R}_{ij}\cdot\mathbf{r}_{ij}}-|\mathbf{R}_{ij}|\right)^2 \\ &= \frac{k}{2}\left(\frac{\mathbf{R}_{ij}\cdot\mathbf{r}_{ij}}{|\mathbf{R}_{ij}|}\right)^2+O(|\mathbf{r}_{ij}|^3), \end{aligned} \tag{4}$$

where $\mathbf{R}_{ij} \equiv \mathbf{R}_i - \mathbf{R}_j$. The large scale motion is then predicted from the slowest nonzero harmonic mode of $V(\{\mathbf{r}_i\}) = \sum_{i<j} E_{ij}$ after accounting for the quadratic order of the expansion only.[16,17,39,52,55–83] The diagonalization of a $3N_s \times 3N_s$ matrix is required for the computation of the harmonic modes, which is performed using EISPACK routines.[85,86]

ANM provides only the mode of the fluctuation and not the amplitude; therefore, d_{cut} is not a natural parameter. In fact, if the lowest eigenvector of the Hessian is denoted as $\mathbf{u}$, which is normalized so that $|\mathbf{u}| = 1$, then the fluctuation is $\mathbf{r} = \epsilon\mathbf{u}$, where ϵ is a very small but arbitrary amplitude. The inter-residue distance is considered to be invariant if and only if

$$||\mathbf{R}_{ij}+\mathbf{r}_{ij}|-|\mathbf{R}_{ij}|| \simeq \epsilon\frac{\mathbf{R}_{ij}\cdot\mathbf{u}_{ij}}{|\mathbf{R}_{ij}|} < d_{\text{cut}}. \tag{5}$$

where $\mathbf{u}_{ij} \equiv \mathbf{u}_i - \mathbf{u}_j$ and again only the leading order of ϵ is taken into account. Therefore, it is more convenient to define a dimensionless parameter $\delta_{\text{cut}} \equiv d_{\text{cut}}/\epsilon$ so that the inter-residue distance is considered invariant if and only if

$$\frac{\mathbf{R}_{ij}\cdot\mathbf{u}_{ij}}{|\mathbf{R}_{ij}|} < \delta_{\text{cut}}. \tag{6}$$

RESULTS

Maximal rigid domains versus all rigid domains

The number of maximal rigid domains is much smaller than that of all possible rigid domains. As an extreme example, if d_{cut} is so large that all the inter-residue distances are considered as invariant, then there is only one maximal rigid domain corresponding to the whole protein structure, whereas there are a total of $2^{N_s}-1$ rigid domains if the subsets are considered as well, where N_s is the protein size. As a more realistic and concrete example, let us consider the human hemoglobin studied in Ref. 9, where the oxy (1HHO) and deoxy (2HHB) forms were compared with $d_{\text{cut}} = 0.30$, and the total number of rigid domains were reported according to their sizes. It has been concluded from the huge number of such domains that the exact counting is not feasible in general. Here we list the maximal rigid domains according to their sizes and compare with the number of all the rigid domains in Table I. We indeed see that the number of maximal rigid domains is much smaller than that of all the rigid domains, and that the exact counting of all the maximal rigid domains is indeed feasible for many proteins, as will be seen in the examples to follow.

Table I
The Number of Maximal Rigid Domains and all Rigid Domains for Human Hemoglobin

Size	Maximal	All
2	0	643
3	2	5341
4	2	29121
5	1	114643
6	0	343572
7	0	808298
8	3	1520258
9	11	2311635
10	19	2861660
11	26	2895704
12	26	2398436
13	12	1623937
14	6	894883
15	3	398055
16	7	141005
17	22	38937
18	37	8098
19	19	1196
20	12	112
21	5	5
Total	213	16395539

The number of maximal rigid domains and all rigid domains of human hemoglobin are obtained from the oxy (1HHO) and deoxy (2HHB) forms with $d_{\text{cut}} = 0.30$, and the results are listed according to their sizes for comparison.

Set of all the maximal rigid domains compared with minimal set of rigid domains with maximal coverage

We illustrate the utility of our method by testing it on the protein LsrB, the quorum sensing receptor of *Salmonella typhimurium*,[87] and the adenylate kinase (AK) of *Escherichia coli*. LsrB and *E. Coli* AK are proteins of chain lengths 314 and 214, respectively. LsrB is a periplasmic protein of the bacterium *S. typhimurium*, and it undergoes conformational transition upon the binding of the signal molecule. The structures of both the unbound (1TM2) and bound (1TJY) forms are deposited in the PDB, shown in Figure 2. Adenylate kinase of *E. coli* also

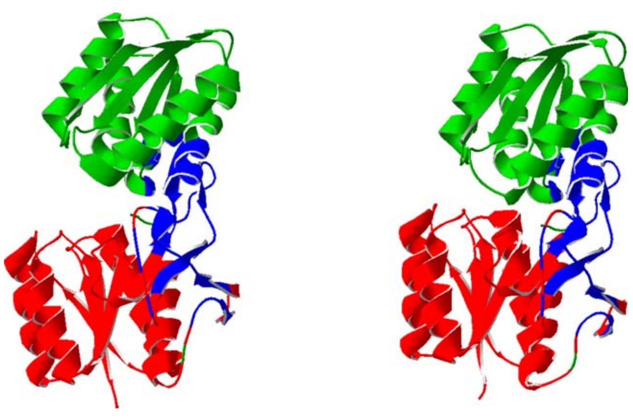

Figure 2

The open and closed conformations of the LsrB. Nonoverlapping parts of domain 1 and domain 2 are colored in red and green, respectively. The hinge region is colored blue. The domains were selected with the exact procedure with $d_{\mathrm{cut}} = 3.00$ Å, without postprocessing. These domains cover the entire structure. The figure was generated with PyMol. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

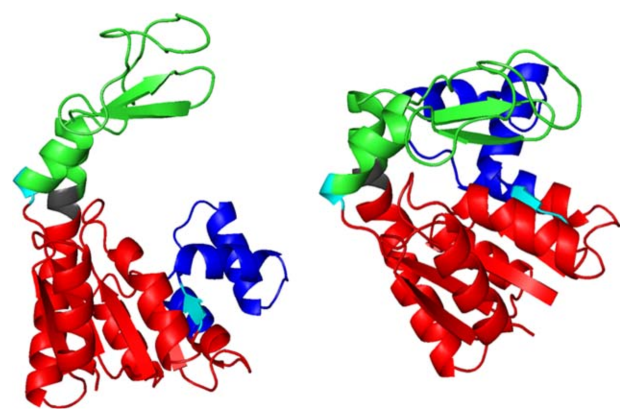

Figure 3

The open and closed conformations of the adenylate kinase of *Escherichia coli.* Nonoverlapping parts of domain 1, domain 2, and domain 3, are colored in red, green, and blue, respectively. The hinge region and the three uncovered residues are colored in cyan and dark gray. The domains were selected with the iterative procedure with $d_{\mathrm{cut}} = 2.5$ Å after postprocessing. The figure was generated with PyMol. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

undergoes conformational change from open to closed form upon binding of adenosine monophosphate (AMP) and adenosine diphosphate (ADP).[56,83,88–90] Both open (4AKE) and closed (2ECK) forms are deposited in PDB, shown in Figure 3. LsrB is considered to consist of two domains,[87] whereas *E. coli* AK is considered to consist of three domains.[56,90] However, the total numbers of maximal rigid domains are much larger than two or three for most values of d_{cut}, as discussed in the previous section. The numbers of maximal rigid domains for both of these proteins are plotted in Figure 4 as functions of d_{cut}, with an interval of 0.25 Å.

The graphs show a general trend toward a decreasing number of maximal rigid domains as the value of d_{cut} increases, but whereas it is a monotonically decreasing function in the case of AK, that of LsrB shows some fluctuations especially for small values of d_{cut}, suggesting that there are fragmented short-scale inter-residue motions scattered throughout the protein. Eventually the number of rigid domains of LsrB becomes one at $d_{\mathrm{cut}} = 8.00$ Å, where the whole structure is regarded as a rigid domain. For AK, this happens at a much larger value of $d_{\mathrm{cut}} = 24.75$ Å, which is due to the fact that the conformational change of AK is much larger than that of LsrB, which is clear from comparing Figures 2 and 3. We see that the d_{cut} value where the whole structure becomes a rigid domain is the effective scale of the conformational change: The conformational change of LsrB becomes invisible at the resolution of 8.00 Å, whereas that of AK is so large that it is visible unless the resolution is worse than 24.75 Å.

In contrast to the total number of maximal rigid domains, the number of rigid domains that cover the whole protein with minimal overlap, found approximately with the iterative algorithm, is more robust with respect to the value of d_{cut}, as plotted with open circles in Figures 5 and 6 for LsrB and AK, respectively. The graph for LsrB exhibits a plateau at $n = 3$ for 2.75 Å $\leq d_{\mathrm{cut}} \leq 6.5$ Å and 7.0 Å $\leq d_{\mathrm{cut}} \leq 7.5$ Å, and that for AK also has a plateau at $n = 3$ for $d_{\mathrm{cut}} \geq 4.75$ Å. However, we note that in the case of LsrB, the three domains

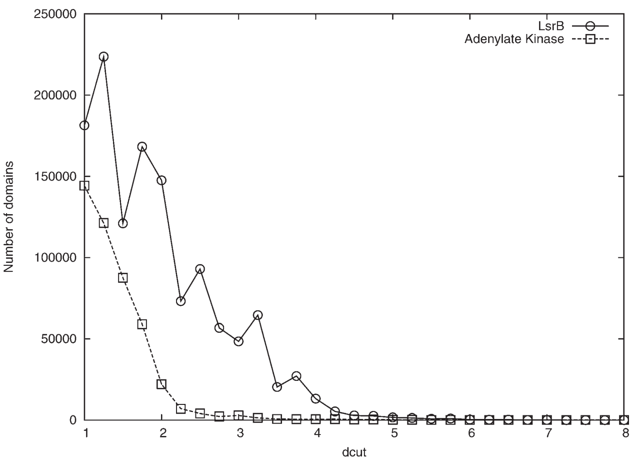

Figure 4

Numbers of maximal rigid domains as functions of d_{cut}. The numbers of the domains become one for $d_{\mathrm{cut}} \geq 8.00$ Å for LsrB, and for $d_{\mathrm{cut}} \geq 24.75$ Å for AK of *E. coli* (out of range of the figure).

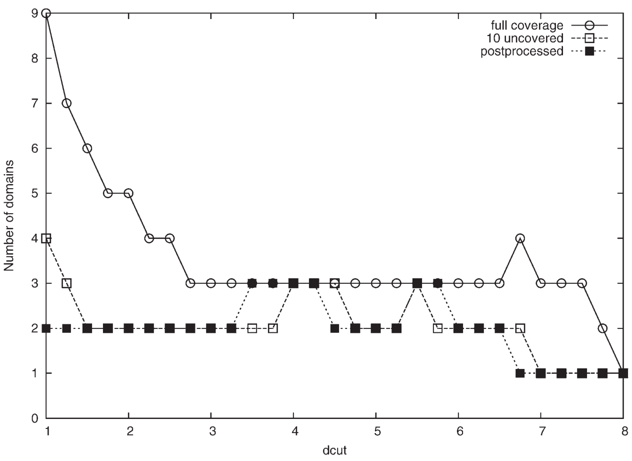

Figure 5

Number of rigid domains with maximal coverage and minimal overlap as a function of d_{cut}, for LsrB. The open circles and open squares denote the results obtained after requiring the full coverage and allowing 10 residues uncovered, respectively, in the absence of postprocessing. The filled squares denote the result obtained after requiring the full coverage, with postprocessing.

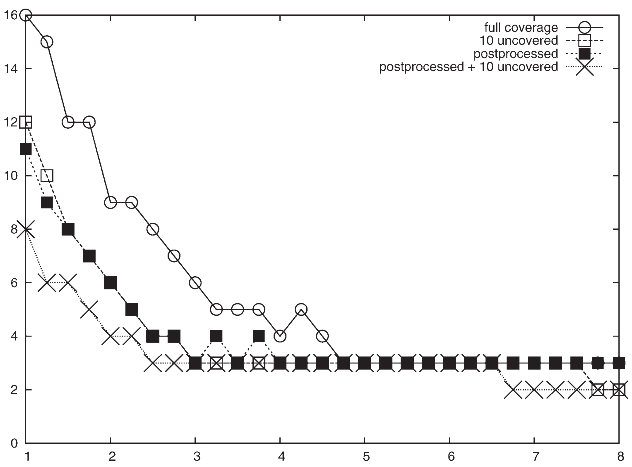

Figure 6

Number of rigid domains with maximal coverage and minimal overlap as a function of d_{cut}, for AK of *E. coli*. The open circles and open squares denote the results obtained after requiring the full coverage and allowing 10 residues uncovered, respectively, in the absence of postprocessing. The filled squares and the crosses denote the results obtained after requiring the full coverage and allowing 10 residues uncovered, respectively, with postprocessing.

selected for these values of cutoff are quite redundant, as can be seen in Figure 7 where the result for $d_{\text{cut}} = 3.00$ Å is shown as an example. As can be seen from the figure, the third domain is almost identical to the first domain, the size of their overlap being 185 residues. Compared to the sizes of the first and the third domains, which are 203 and 194, respectively, the size of the overlapping region forms 91% of the first domain and 95% and the third domain. In fact, the first and the second domains cover 98% of the whole protein, leaving only 5 residues uncovered at positions 1, 40–41, 269, and 274, as shown in Figure 7. The redundant third domain was required only to cover these gaps.

One way to remove the redundancy of the domains is to drop the overly strict requirement of full coverage. If we allow 10 residues to be left uncovered, being about 3% of the protein chain, only two domains are selected for 1.50 Å $\leq d_{\text{cut}} \leq$ 3.75 Å, 4.75 Å $\leq d_{\text{cut}} \leq$ 5.25 Å and 5.75 Å $\leq d_{\text{cut}} \leq$ 6.75 Å, as shown in Figure 5 with open squares. In fact, with $n = 2$, the uncovered region does not exceed 10% of the whole protein chain for all the values of d_{cut} considered here, as shown in Supporting Information Figure S4 (b) with filled circles.

Postprocessing is an alternative way to prevent redundant domains. As shown in Figure 7, the residues uncovered by the first two domains form short gaps interspersed throughout the protein structure, most of which can be removed by postprocessing (see Methods). Indeed, postprocessing yields wide ranges of d_{cut} values where $n = 2$, at $d_{\text{cut}} \leq$ 3.25 Å, 4.50 Å $\leq d_{\text{cut}} \leq$ 5.25 Å, and 6.00 Å $\leq d_{\text{cut}} \leq$ 6.50 Å, even if we require the full coverage (Fig. 5, filled squares).

In contrast to that of LsrB, the plateau at $n = 3$ for AK is quite robust. The plateau remains even if we allow 10 residues to be uncovered, comprising nearly 5% of the

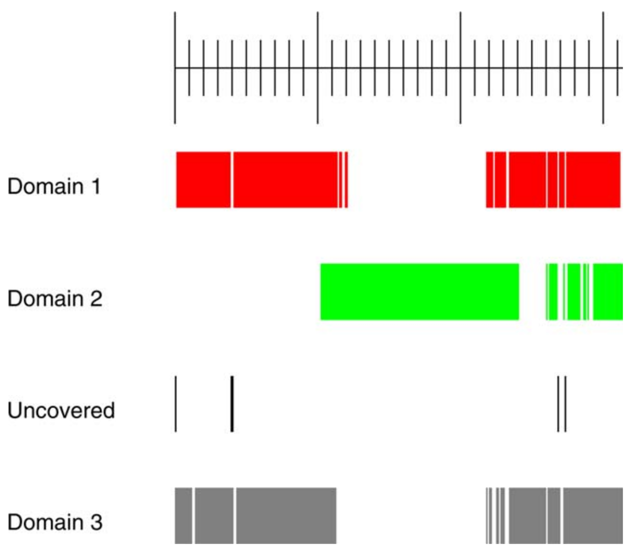

Figure 7

Maximal rigid domains of LsrB selected with the approximate iterative algorithm for $d_{\text{cut}} = 3.00$ Å without postprocessing. The residues uncovered by the first two domains are shown in black. The third domain, which is almost identical to the first domain, is required only to cover these gaps. Here and in the figures to follow, the rulers at the top have small tick marks at 10 residues interval, and large tick marks at 100 residues interval. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

protein chain (Fig. 6, open squares). The plateau is robust also with respect to postprocessing (Fig. 6, filled squares). The plateau remains even when postprocessing is performed and ten residues are allowed to be uncovered at the same time (Fig. 6, crosses). The result indicates that the protein AK should be considered as consisting of three distinct domains. The three domains that cover most of the protein for $d_{\mathrm{cut}} = 2.50$ Å, obtained without postprocessing, are shown in Supporting Information Figure S1, along with the twenty uncovered residues scattered throughout the structure.

The two domains for LsrB and the three domains for AK are also shown along the protein structures in Figures 2 and 3 respectively. In Figure 2, the domains for LsrB are selected for $d_{\mathrm{cut}} = 3.00$ Å with the exact procedure, which do not deviate much from those obtained from the iterative procedure, and will be discussed again in the next section. The domains shown in Figure 3 are obtained for $d_{\mathrm{cut}} = 2.50$ Å after postprocessing, which will also be discussed in more detail later.

Robustness with respect to the selection procedure

When the exact selection procedure is employed (See Methods section), two maximal rigid domains that cover the whole protein chain of LsrB can be found for certain values of d_{cut}, even without postprocessing. The result for $d_{\mathrm{cut}} = 3.00$ Å is shown in Supporting Information Figure S2 as an example, where we see that the difference from the results obtained with the iterative procedure is quite small. These domains and their overlapping region are also shown in Figure 2 along the protein structure.

To quantitatively assess the difference between the exact and the iterative selection procedure, we compared the domains selected with these two procedures for 2.00 Å $\leq d_{\mathrm{cut}} \leq$ 8.00 Å with interval of 0.25 Å in Supporting Information Figure S3, where we plotted the number of residues not common to the equivalent counterparts (method sensitivity), along with the size of the two domains.[2] We see that the second domains selected by the two methods are more or less identical. On the other hand, the first domain is less robust with respect to the selection method. This is due to the fact that in the iterative procedure, the selection criterion for the first domain is its large size regardless of the size of the overlapping region. However, we see that the result is robust for most values of d_{cut}, except around the value of 4.25 Å. The reasonable robustness of the result in regard to the selection method justifies the approximate iterative selection procedure that is computationally much more efficient, since iterative and exact methods require computational times that are proportional to nN and N^n, respectively.

Dependence on d_{cut}

The dependence of the selected domains of LsrB and AK on the value of d_{cut} was examined for the range of d_{cut} values considered in the previous section, in the absence of postprocessing. The two domains and three domains were examined for LsrB and AK, respectively. The sizes of the selected domains, as well the number of their common residues and those not belonging to either of these domains, are plotted in Supporting Information Figure S4. The results obtained from both the exact and the iterative methods are displayed for LsrB [Supporting Information Fig. S4(a,b)].

We see that the sizes of the selected domains as well as their common residues increase with increasing values of d_{cut} for both proteins, but the size of the second domain of AK is rather robust with respect to the value of d_{cut}. For LsrB, two domains that cover the whole protein can be found using the exact procedure for $d_{\mathrm{cut}} > 2.50$ Å [Supporting Information Fig. S4(a)]. When the two domains are selected using the approximate iterative method where the largest domain is found first and fixed, uncovered regions of sizes up to 10% of the sequence length appears [Supporting Information Fig. S4(b)]. However, except for 4.00 Å $\leq d_{\mathrm{cut}} \leq$ 4.50 Å, the domains obtained from the iterative method agrees well with those obtained with the exact method (Supporting Information Fig. S3), and consequently the size of the uncovered region does not exceed 4% [Supporting Information Fig. S4(b)]. For AK, the three domains selected from the iterative procedure covers the entire protein chain for $d_{\mathrm{cut}} \geq 4.75$ Å [Supporting Information Fig. S4(c)].

Because the domain size does not contain information on the actual residues belonging to the domain, we also compared a domain and its equivalent counterpart obtained with a d_{cut} value 0.25 Å smaller than the current one, and plotted the number of residues that are not common to these two equivalent domains as the function of d_{cut} (Supporting Information Fig. S4, cutoff sensitivity). We see that domains with a 0.25 Å difference in d_{cut} overlap reasonably well. Therefore, we see that the effect of increasing d_{cut} is to increase the size the selected domains without an abrupt change in their numbers or identities, when a few residues are allowed to be uncovered. Because the number of common residues between the selected domains also increases with increasing value of d_{cut}, it can be considered as a free parameter that can be chosen according to the desired size of the hinge region. For example, for $d_{\mathrm{cut}} = 3.00$ Å, we obtain the pair of maximal rigid domains that cover the whole

[2] When the pair of domains with maximal coverage and minimal overlap is selected using the exact procedure, the ordering of these two domains is arbitrary, in contrast to the iterative method where the largest maximal rigid domain is selected first. Therefore, for consistency of comparison, we ordered the domains so that the larger one is called the first domain.

sequence with an overlap size of 164 residues without postprocessing, as shown in Supporting Information Figure S2, whereas the size of the overlap region is 8 residues for $d_{\mathrm{cut}} = 1.25$ Å with postprocessing, as shown in Supporting Information Figure S5. When one performs a molecular dynamics simulation with fixed conformations of the rigid domains covering the chain, the size of the hinge region is directly proportional to the degrees of freedom, which is to be chosen by making a compromise between computational costs and accuracy, by controlling the value of d_{cut}.

Comparison with other methods

Our result can be compared with those of RigidFinder[10] and DynDom,[15] which provide web servers that can be used easily. RigidFinder is a heuristic method that divides a protein structure into non-overlapping rigid regions. The largest rigid region is found by an approximate method and removed from the structure, and the computation is repeated until most of the conformation is covered. In contrast to the distance-based definition of rigid domains used in both DAGR and RigidFinder, DynDom defines rigid domains by clustering the residues according to the rotational vectors of their displacements after superposing two conformations. Therefore d_{cut} is needed only for DAGR and RigidFinder. Larger values of d_{cut} resulted in larger sizes of Domain 1 both for DAGR and RigidFinder (see Supporting Information Figs. S3 and S4 for DAGR data in the absence of postprocessing. The data for RigidFinder are not shown). DynDom did not allow any free parameter to be put in by users, so we adjusted the value of d_{cut} to 1.25 Å for LsrB in order to obtain the best agreement between RigidFinder and DynDom. The DAGR results obtained after postprocessing are compared with those obtained with RigidFinder and DynDom in Supporting Information Figure S5. We see that the size of the first domain (red) found by DAGR is 173, which is slightly larger than that obtained by RigidFinder (169), indicating that the largest rigid domain found by exact algorithm of DAGR is missed by RigidFinder. Because RigidFinder partitions the protein into non-overlapping domains, it also misses information on the overlap regions of rigid domains. DynDom also reports on hinge residues that are in accordance with our result for this example. However, definition of both the rigid domains and the hinge residues are different from ours in that they are defined in terms of local rotation vectors. In DynDom, the hinges are defined as residues near the inter-domain boundary with somewhat intermediate values of the rotation vectors.[15] Our definition of the hinge region as comprising the common residues between the maximal rigid domains, provides more useful information on the property of inter-domain motion, as will be seen in the next example of *E. coli* AK. In fact, no common residues will exist on the inter-domain boundary if the boundary does not act as a hinge of a bending motion. Also, our method has an advantage that the size of the overlapping region can be controlled by d_{cut} as discussed in the previous section. Therefore, a hinge is not necessarily localized near a boundary, as can also be seen in Supporting Information Figure S2 where the hinge residues are shown in blue for $d_{\mathrm{cut}} = 3.00$ Å in the absence of postprocessing.

The result for AK after postprocessing is shown in Figure 8 along with those obtained from RigidFinder and DynDom, where we set $d_{\mathrm{cut}} = 2.5$ Å, the value used in Ref. [10] in a benchmark test. For simplicity of the figure, the first three domains are shown at once, where the first domain (CORE) is colored red, and the part of the second domain (LID) not covered by the first is colored green, and the part of the third domain (AMP-binding) not covered by the previous domains are colored blue. This method of output is the one used in RigidFinder.[10] The overlapping regions of these three domains are also shown [Hinge (1–3)], so the full ranges of the second and the third domains can be easily reconstructed. We see that the first three domains cover most of the protein, leaving only three residues uncovered at positions 168–170. These three domains are also predicted from RigidFinder and DynDom. These domains are also extensively discussed in the literature, and called CORE, LID, and AMP-binding domains, respectively.[90] The non-overlapping and overlapping parts of these three domains are also shown along the protein conformation in Figure 3.

In contrast to the case of LsrB, the DAGR result for AK has some clearly distinguishing features compared to those obtained from other methods. There are two boundaries between the CORE and the AMP-binding domains along the sequence, and in contrast to the case of LsrB studied in the previous sections, only one of them has overlapping region. This is due to the fact that the motions of the protein conformation with respect to these boundaries exhibit distinct behaviors. The segments of the CORE domain flanking the AMP-binding domain were extracted from both open and closed conformations, superposed with minimal root mean square deviation of their coordinates, and shown with AMP-binding domain in Supporting Information Figure S6. We see that indeed the AMP-binding domain undergoes bending with respect to the hinge residues. On the other hand, the other boundary lies at the opposite side of the hinge where these two domains close in toward each other. This information is hard to obtain from the hinge reported from DynDom, since hinge residues appear at each of the boundary by construction.

For the case of the CORE domain versus LID domain, one of the boundary forms a hinge, but the other boundary undergoes relatively smooth deformation instead of a sharp change (Supporting Information Fig.

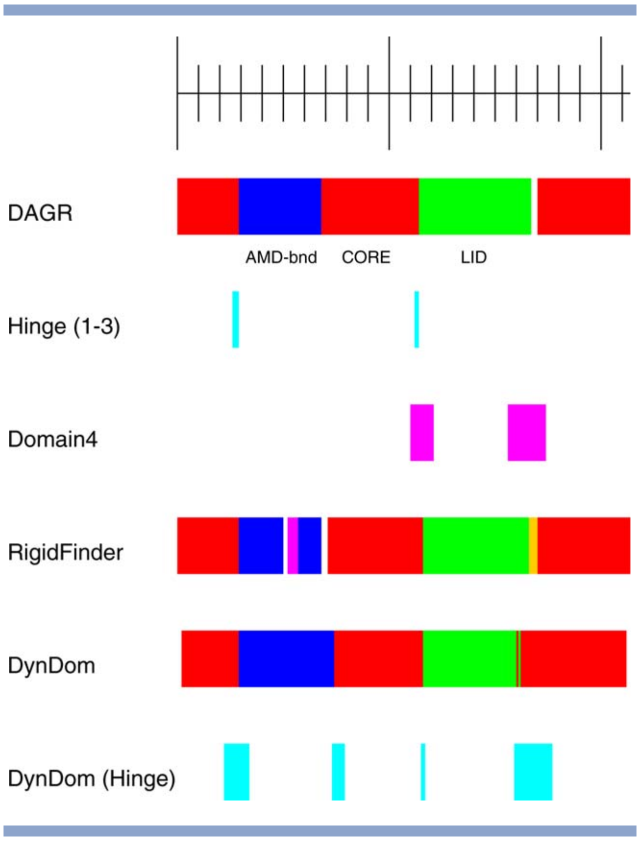

Figure 8

The three maximal rigid domains with minimal overlap and maximal coverage, obtained after postprocessing, denoted as CORE, LID, and AMP-bnd, following the literature. The common residues are shown and are denoted as Hinge. The fourth domain required to cover the whole protein chain is also shown, along with the results from RigidFinder, DynDom, and the hinge region reported from DynDom. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

S7). This fact is indicated by the presence of uncovered residues at this boundary, and that an additional rigid domain has to be selected in order to cover this region, denoted as domain 4 in Figure 8. The DynDom also reports a relatively large hinge at this region, supporting this fact. We get similar large hinge regions if the 4th domain is included, as is clear in the figure.

There is a counterpart of this 4th domain also in the case of RigidFinder, but it should be noted that there is an additional rigid domain of size 5 at positions 53–57, separating the CORE domain. Because our result shows that the CORE rigid domain forms a contiguous stretch along the sequence without a break, the appearance of the additional domain seems to be an artifact of using an approximate algorithm for finding maximal rigid domains. Also note that in the RigidFinder, five residues are still uncovered at positions 51–52 and 69–71 even with the five rigid domains reported, whereas the four maximal rigid domains reported by DAGR fully cover the structure.

We see from Figure 6 that when 10 residues are allowed to be uncovered, two domains are enough cover the protein chain for $d_{\mathrm{cut}} \geq 6.75$ Å and $d_{\mathrm{cut}} \geq 7.75$ Å with and without postprocessing, respectively. The result for $d_{\mathrm{cut}} = 8.00$ Å after postprocessing is shown in Supporting Information Figure S8 as an example, where these two domains that cover most of the protein are shown to be the CORE and LID domains, leaving only nine residues uncovered at the center of what used to be the AMP-binding domain. This result corresponds to a low-resolution description of the conformational change where the relatively small movement of the AMP-binding domain relative to the CORE domain is neglected and the AMP-binding domain is effectively absorbed into the CORE domain. Only the large movement of the LID domain with respect to the CORE domain is taken into account at this scale (Supporting Information Fig. S8).

We also applied the current method to the set of proteins used in Table I in Ref. [10]. Some of the results are shown in Table II, which are sorted according to the number of maximal rigid domains *N*. The *E. coli* adenylate kinase studied in the previous sections is included in this set, which is denoted simply as phosphotransferase. Structures with different sequences for malate dehydrogenase were compared in Ref. [10], which is not included here since the result may strongly depend on how we align the two proteins. The same values of d_{cut} were used as in Table I in Ref. [10]. Postprocessing was also performed and the same level of coverage was required as in the case of RigidFinder for a fair comparison. We see that the number of selected domains with minimal overlap and maximal coverage, *n*, are in reasonable agreement with the number of rigid domains found from RigidFinder and DynDom, denoted as n_{RF} and n_{DD}. On the other hand, we find that the maximum size of the rigid domains found with the current method, $L_{\max}$, is always larger than that obtained with RigidFinder, $L_{\max}^{\mathrm{RF}}$, due to the exact nature of our computation. It is to be noted that the comparing the maximum size of the rigid domain with that of DynDom is not very meaningful since their definition of a rigid domain is different from ours.

Prediction of rigid domains

We also combined DAGR with ANM to predict rigid domains. The total number of predicted rigid domains as a function of δ_{cut} exhibits behavior similar to that of actual rigid domains as a function of d_{cut} in that it decreases with an increasing cutoff value, as shown in Supporting Information Figures S9 and S10 for LsrB and AK, respectively.

Rigid domains with maximal coverage and minimal overlap tend to fragment even after postprocessing, especially for larger values of δ_{cut}, as shown in Supporting Information Figure S11 for the case of the first domain

Table II
Test Results for Various Proteins

Protein name	Size	N	n	n_{RF}	n_{DD}	$L_{\max}$	$L_{\max}^{\mathrm{RF}}$	T_{gen}	T_{proc}
Cro repressor	60	59	2	2	–	53	53	$<10^{-6}$ s	$<10^{-6}$ s
Calmodulin	138	121	3	3	2	67	65	$<10^{-6}$ s	$<10^{-6}$ s
HIV-1 protease	99	276	2	2	2	92	70	$<10^{-6}$ s	$<10^{-6}$ s
Antigen 85C	280	437	3	3	2	260	257	$<10^{-6}$ s	$<10^{-6}$ s
S100A6	89	1031	6	7	–	40	28	$<10^{-6}$ s	$<10^{-6}$ s
Phosphotransferase	214	4015	3	5	3	119	118	$<10^{-6}$ s	0.01 s
Bungarotoxin	74	5280	5	4	–	28	19	0.01*s*	$<10^{-6}$ s
DNA polymerase beta	326	191521	3	3	3	167	164	0.5 s	0.05 s
Bacteriorhodopsin	170	201122	6	5	–	108	93	0.2 s	0.03 s
Pyruvate phosphate dikinase	872	433605	10	10	3	391	391	3 s	0.3 s
T7 RNA polymerase	843	466644	8	8	2	556	487	2 s	0.8 s
Aspartate aminotransferase	401	4797848	4	3	2	303	285	18 s	1 s
Alcohol dehydrogenase	374	56393147	2	4	2	236	214	72 s	130 s

Benchmark test results for the proteins of Ref. 10. See Table I of the reference for details such as PDB ID and the values of d_{cut}. N is the total number of maximal rigid domains and n is the number of the selected domains that cover most of the protein chain with minimal overlap. The number of domains found by RigidFinder[10] and DynDom[15] are denoted as n_{RF} and n_{DD}, respectively. $L_{\max}$ and $L_{\max}^{\mathrm{RF}}$ are the sizes of the largest rigid domains found by the current method and RigidFinder, respectively. The computer time required for the generation of all the maximal rigid domains is denoted as T_{gen}, and that for the postprocessing and the selection of the n maximal rigid domains is denoted as T_{proc}.

of LsrB, which was predicted from the open conformation. This problem, which is not seen in cases of actual rigid domains computed from two conformations, is probably due to errors in the predicted motion. To avoid this problem, we added another selection criterion for the case of domain prediction, in which the domain containing the longest stretch of contiguous residues in the nonoverlapping region is selected before considering the other criteria, as shown in Supporting Information Figure S12. For the computation of actual rigid domains from two conformations, the result does not change for most values of d_{cut}, except for a trivial reordering of the domains, but the number of selected domains becomes slightly more robust (data not shown). As in the case of actual rigid domains computed from two conformations, the number of selected rigid domains is much less than the total number of maximal rigid domains, and approximately robust with respect to the value of δ_{cut}, as shown in Supporting Information Figure S13.

The predicted maximal rigid domains are shown in Figures 9 and 10 for LsrB and *E. coli* AK, respectively, for two selected values of δ_{cut}, and compared with the actual rigid domains computed from two conformations for two selected values of d_{cut} as well as the results from previous methods for rigid domain prediction, StoneHinge[44] and HingeProt.[45] StoneHinge consists of two methods, StoneHingeP and StoneHingeD. StoneHingeP is based on the FIRST method,[22] which counts various constraints in the protein structure and identifies overconstrained and underconstrained regions as rigid and flexible, respectively.[21–28] StoneHingeD is based on DomDecomp, which utilizes GNM.[41] HingeProt also utilizes GNM for domain parsing, and ANM for predicting the motion.[45] Only the HingeProt result from the slowest mode is shown here, although HingeProt also reports the result from the next slowest mode.

We note that it is easier to predict the rigid domain using ENM from the open conformation, where rigid domains are well separated, than from the closed conformation, which also can be seen from the DAGR results. For example, the predicted domains for LsrB tend to become slightly fragmented when the closed conformation is used (Fig. 9). The prediction results from StoneHingeD and HingeProt are more robust with respect to the conformation used for the prediction, which seems to be due to the additional filter used in these methods for assigning a domain, such as the continuity of β-sheets[41] or the compactness score.[45] Note, however, that StoneHingeD predicts three domains for LsrB instead of two from both the open and closed conformations (Fig. 9). For *E. coli* AK, all the methods predict two domains, most of them failing to assign the AMP binding domain as an independent domain. For unknown reasons, StoneHingeD includes this region in the LID domain and DAGR assigns it into both the LID and CORE domains simultaneously when the closed conformation is used for prediction (Fig. 10).

Overall, the domains predicted by combining DAGR with ANM agree reasonably well with the actual rigid domains as well as those from other methods (Figs. 9 and 10), but DAGR provides additional information on the overlap between rigid domains. In contrast, both StongeHingeD and HingeProt are methods that partition the protein into nonoverlapping domains, and the hinge residues are defined as either pairs of residues (StoneHingeD) or one residue (HingeProt) at each interdomain boundary, and thus are extremely localized. In addition, although StoneHingeP shares the feature with DAGR that the hinge region is not necessarily localized,

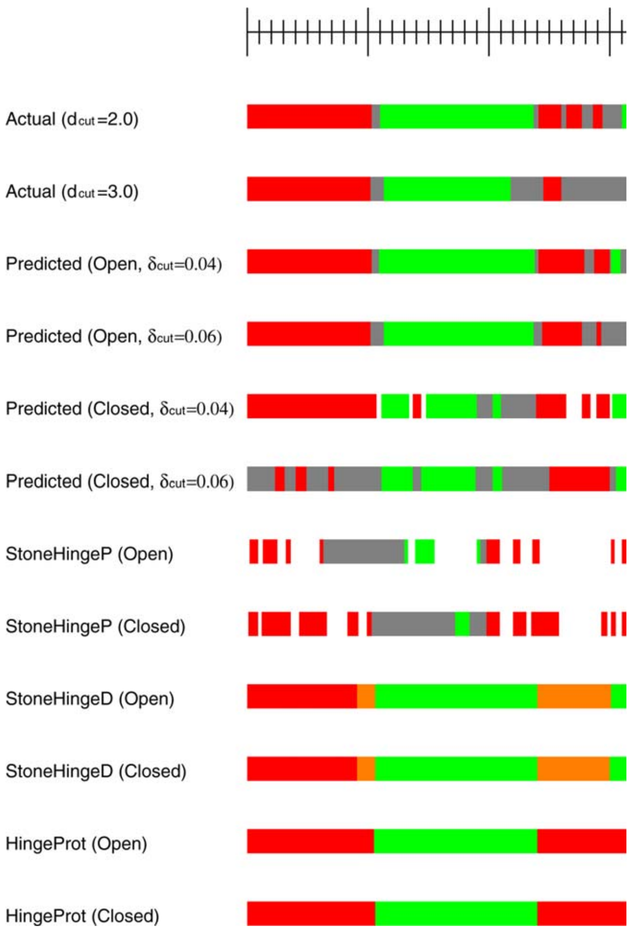

Figure 9

The rigid domains of LsrB computed using DAGR from two conformations for d_{cut} = 2.0Å and d_{cut} = 3.0Å, as well as prediction results obtained by applying DAGR and ANM to open and closed conformations, for δ_{cut} = 0.04 and δ_{cut} = 0.06. The nonoverlapping part of the rigid domains are shown in color and the overlapping region is shown in grey. Prediction results of StoneHingeP, StoneHingeD, and HingeProt are also shown. Hinge residues reported by StoneHingeP are shown as grey region. For StoneHingeD and HingeProt, the hinge residues are defined as one or two residues at the inter-domain boundary. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

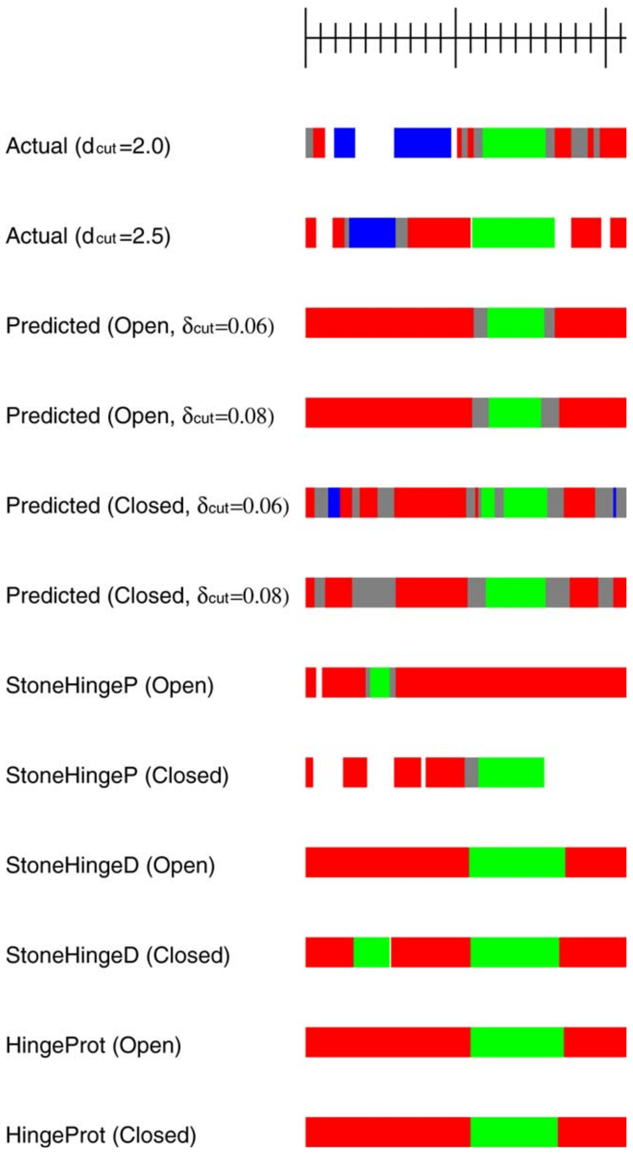

Figure 10

The rigid domains of *E. coli* AK computed using DAGR from two conformations for d_{cut} = 2.0Å and d_{cut} = 2.5Å, as well as prediction results obtained by applying DAGR and ANM to open and closed conformations, for δ_{cut} = 0.06 and δ_{cut} = 0.08. The notations are the same as in Figure 9.

the hinge region in StoneHingeP is a flexible region between relatively rigid domains, which is conceptually different from that of DAGR in which the hinge region is defined as the overlapping region between the rigid domains. Given that these two methods are based on distinct conceptual frameworks, we consider the information provided by the two methods to be complementary.

To summarize, a simple combination of ANM with DAGR without additional sophisticated filters is able to yield reasonably good prediction results. This finding demonstrates the predictive power of ANM as well as the utility of the DAGR method for predicting the hinge region as the overlap of rigid domains.

Computation time

When computing actual rigid domains of LsrB from two conformations, with d_{cut} = 3.00 Å, the central processing unit (CPU) time for generating 48,421 rigid domains is of the order of 0.01 seconds using an Intel i3 CPU (3.30 GHz). The time for selecting the final two or three domains covering the protein is also of the same order if we choose the approximate iterative method, whereas it takes about 30 seconds if we use the exact method. In the benchmark test results in Table II, the time for generating all the maximal rigid domains is denoted as T_{gen} and that for postprocessing and selecting the domains with maximal coverage and minimal overlap, is denoted as T_{proc}. We see that when the number of maximal rigid domains is $<5\times10^5$, it takes only a few seconds to generate the domains and less than a second to process and select the domains that cover the whole protein with minimal overlap. For aspartate aminotransferase, whose total number of maximal rigid domains is around 5×10^6, T_{gen} and T_{proc} increase to 18 seconds and 1 second, respectively. The number of maximal rigid domains for alcohol dehydrogenase was about 6×10^7, which could be counted in about 1 min, but T_{proc} is about 2 minutes, most of which is due to postprocessing. Without postprocessing, the time for selecting the final domains takes less than ten seconds. For the remaining four proteins in Table I of Ref. [10], the number of maximal rigid domains exceeded 10^8, in which case even the maximal rigid domains could not be generated in a reasonable amount of time. Because the number of maximal rigid domains is not known beforehand, the most reasonable way to utilize the current method is to set the limit on the number of maximal rigid domains to be around 10^8, and stop the computation when this limit is reached. Then the user should either increase the value of d_{cut} with an expectation that the number of maximal rigid domains will decrease, or resort to other heuristic methods to find approximate rigid domains.

In the case of rigid domain prediction, additional time is required for diagonalizing a $3N_s\times3N_s$ matrix, which is expected to scale as $O(N_s^3)$.[86] Including a few seconds of overhead for memory allocation, it takes about 2 seconds for Bungtoroxin with $N_s = 74$ to about 4.5 minutes for Pyruvate phosphate dikinase with $N_s = 872$.

Web server

We implemented our algorithm on a web server, which is available at http://dna.ssu.ac.kr/index.php?pid=program.

DISCUSSION

The advantage of the current method, DAGR, is that since it generates all the maximal rigid domains, one can select the maximal rigid domains with desired properties such as minimal overlap and maximal coverage, or maximal length of the longest stretch of contiguous residues in the non-overlapping region, along with flexibility as to how much of the protein can be left uncovered by these domains. Furthermore, the overlap of the maximal rigid domains provides us with valuable information on the hinge region, which is missed with methods that partition the protein into non-overlapping domains. The existence of the free parameter d_{cut} provides an additional convenience in that a user can control the size of the hinge region.

The cost of exhaustive enumeration of all the maximal rigid domains may become formidable both in terms of CPU time and memory requirements, when there are too many maximal rigid domains for a given value of d_{cut}, but it is possible to overcome the size problem by using parallelization. The Bron-Kerbosch algorithm can be easily adapted for parallel computation, so that a number of computational nodes share the burden of computation.[91] The key point is to implement dynamical load balancing so that the work is distributed fairly and no node becomes idle while others are working hard.

To predict rigid domains using DAGR, an alternate conformation has to be generated. Although we employed ANM to perform the task, any method of protein motion prediction can be used. In fact, the prediction results from combination of DAGR and ANM show that there are rooms for improvement. Perhaps the performance can be enhanced by replacing ANM with physics-based force fields, based on either all-atom or coarse-grained models.

CONCLUSION

In this article, we have presented a novel method, DAGR, for computing maximal rigid domains based on an algorithm for finding maximal cliques in graph theory. The current method easily generates all the maximal rigid domains present, and selects the maximal rigid domains with minimal overlap with the desired level of coverage, all within several seconds, as long as the actual number of maximal rigid domains does not exceed 5×10^6. The overlap region of the selected maximal rigid domains corresponds to the hinge region, which plays a crucial role in the large-scale movement of the protein. The existence or absence of the overlap reveals the nature of the inter-domain motion, and its size can be controlled for the purpose of molecular dynamics simulations where most of the residues in the rigid domains are to be fixed. We also showed that the method can also be used for predicting the rigid domains from a single conformation, by generating alternative conformation via ANM.

ACKNOWLEDGMENTS

The authors thank Sun Choi and Changbong Hyeon for many insightful discussions.

REFERENCES

1. Jenwitheesuk E, Samudrala R. Virtual screening of HIV-1 protease inhibitors against human cytomegalovirus protease using docking and molecular dynamics. AIDS 2005;19:529–533.
2. Bahar I, Chennubhotla C, Tobi D. Intrinsic dynamics of enzymes in the unbound state and relation to allosteric regulation. Curr Opin Struc Biol 2007;17:633–640.

3. Totrov M, Abagyan R. Flexible ligand docking to multiple receptor conformations: a practical alternative. Curr Opin Struc Biol 2008; 18:178–184.
4. Nilmeier J, Hua L, Coutsias EA, Jacobson MP. Assessing protein loop flexibility by hierarchical monte carlo sampling. J Chem Theory Comput 2011;7:1564–1574.
5. Shin WH, Kim JK, Kim DS, Seok C. GalaxyDock2: protein-ligand docking using beta-complex and global optimization. J Comput Chem 2013;34:2647–2656.
6. Jolley C, Wells SA, Hespenheide BM, Thorpe MF, Fromme P. Docking of photosystem I subunit C using a constrained geometric simulation. J Am Chem Soc 2006;128:8803–8812.
7. Flores S, Echols N, Milburn D, Hespenheide B, Keating K, Lu J, Wells S, Yu EZ, Thorpe M, Gerstein M. The database of macromolecular motions: new features added at the decade mark. Nucleic Acids Res 2006;34:d296–d301.
8. Gohlke H, Thorpe MF. A natural coarse graining for simulating large biomolecular motion. Biophys J 2006;9:2115–2120.
9. Nichols WL, Rose GD, Ten Eyck LF, Zimm BH. Rigid domains in proteins: an algorithmic approach to their identification. Proteins 1995;23:38–48.
10. Abyzov A, Bjornson R, Felipe M, Gerstein M. RigidFinder: a fast and sensitive method to detect rigid blocks in large macromolecular complexes. Proteins 2010;78:309–324.
11. Ekman D, Björklund ÅK, Frey-Skött J, Elofsson A. Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions. J Mol Biol 2005;348:231–243.
12. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. Nucleic Acids Res 2000;28:235–242.
13. Boutonnet NS, Rooman MJ, Wodak SJ. Automatic analysis of protein conformational changes by multiple linkage clustering. J Mol Biol 1995;253:633–647.
14. Wriggers W, Schulten K. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. Proteins 1997;29:1–14.
15. Hayward S, Berendsen HJ. Systematic analysis of domain motions in proteins from conformational change: new results on citrate synthase and t4 lysozyme. Proteins 1998;30:144–154.
16. Hinsen K, Thomas A, Field MJ. Analysis of domain motions in large proteins. Proteins 1999;34:369–382.
17. Yang L, Song G, Jernigan RL. How well can we understand large-scale protein motions using normal modes of elastic network models? Biophys J 2007;93:920–929.
18. Bron C, Kerbosch J. Algorithm 457: finding all cliques of an undirected graph. Commun ACM 1973;16:575–577.
19. Tomita E, Tanaka A, Takahashi H. The worst-case time complexity for generating all maximal cliques and computational experiments. Theor Comput Sci 2006;363:28–42.
20. Koike R, Ota M, Kidera A. Hierarchical description and extensive classification of protein structural changes by motion tree. J.Mol Biol 2014;426:752–762.
21. Thorpe MF, Lei M, Rader AJ, Jacobs DJ, Kuhn LA. Protein flexibility and dynamics using constraint theory. J Mol Graph Modell 2001;19:60–69.
22. Jacobs DJ, Rader AJ, Kuhn LA, Thorpe MF. Protein flexibility predictions using graph theory. Proteins 2001;44:150–165.
23. Jacobs DJ, Dallakyan S, Wood GG, Heckathorne A. Network rigidity at finite temperature: relationships between thermodynamic stability, the nonadditivity of entropy, and cooperativity in molecular systems. Phys Rev E 2003;68:061109.
24. Jacobs DJ, Wood GG. Understanding the α-helix to coil transition in polypeptides using network rigidity: predicting heat and cold denaturation in mixed solvent conditions. Biopolymers 2004;75:1–31.
25. Wells S, Menor S, Hespenheide B, Thorpe MF. Constrained geometric simulation of diffusive motion in proteins. Phys Biol 2005;2: S127–S136.
26. Jacobs DJ, Livesay DR, Hules J, Tasayco ML. Elucidating quantitative stability/flexibility relationships within thioredoxin and its fragments using a distance constraint model. J Mol Biol 2006;358:882–904.
27. Chubynsky MV, Thorpe MF. Algorithms for three-dimensional rigidity analysis and a first order percolation transition. Phys Rev E 2007;76:041135.
28. Chubynsky MV, Hespenheid B, Jacobs DJ, Kuhn LA, Lei M, Menor S, Rader AJ, Thorpe MF, Whiteley W, Zavodszky MI. Constraint theory applied to proteins. Nanotechnol Res J 2008;2:61–72.
29. Lee Y, Choi S, Hyeon C. Mapping the intramolecular signal transduction of G-protein coupled receptors. Proteins 2014;82:727–743.
30. Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. J Comput Biol 1999;6:281–297.
31. Tanay A, Sharan R, Shamir R. Discovering statistically significant biclusters in gene expression data. Bioinformatics 2002;18 (Suppl 1): S136–S144.
32. Sugihara G. Graph theory, homology and food webs. Proc Sym Ap 1984;30:83–101.
33. Day WHE, Sankoff D. Computational complexity of inferring phylogenies by compatibility. Syst Zool 1986;35:224–229.
34. Samudrala R, Moult J. A graph-theoretic algorithm for comparative modeling of protein structure. J Mol Biol 1998;279:287–302.
35. Chen Y, Crippen GM. A novel approach to structural alignment using realistic structural and environmental information. Protein Sci 2005;14:2935–2946.
36. Schmitt S, Kuhn D, Klebe G. A new method to detect related function among proteins independent of sequence and fold homology. J Mol Biol 2002;323:387–406.
37. Spirin V, Mirny LA. Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci USA 2003;100:12123–12128.
38. Rhodes N, Willett P, Calvet A, Dunbar JB, Humblet C. CLIP: similarity searching of 3D databases using clique detection. J Chem Inf Model 2003;43:443–448.
39. Hayward S, Kitao A, Berendsen HJ. Model-free methods of analyzing domain motions in proteins from simulation: a comparison of normal mode analysis and molecular dynamics simulation of lysozyme. Proteins 1997;27:425–437.
40. Hinsen K. Analysis of domain motions by approximate normal mode calculations. Proteins 1998;33:417–429.
41. Kundu S, Sorensen DC, Phillips GN. Automatic domain decomposition of proteins by a gaussian network model. Proteins 2004;57: 725–733.
42. Flores SC, Gerstein MB. FlexOracle: predicting flexible hinges by identification of stable domains. BMC Bioinform 2007;8:215–231.
43. Flores SC, Keating KS, Painter J, Morcos F, Nguyen K, Merritt EA, Kuhn LA, Gerstein MB. HingeMaster: normal mode hinge prediction approach and integration of complementary predictors. Proteins 2008;73:299–319.
44. Keating KS, Flores SC, Gerstein MB, Kuhn LA. StoneHinge: hinge prediction by network analysis of individual protein structures. Protein Sci 2009;18:359–371.
45. Emekli U, Schneidman-Duhovny D, Wolfson HJ, Nussinov R, Haliloglu T. HingeProt: automated prediction of hinges in protein structures. Proteins 2008;70:1219–1227.
46. Arnold GE, Ornstein RL. Molecular dynamics study of time-correlated protein domain motions and molecular flexibility: cytochrome P450BM-3. Biophys J 1997;73:1147–1159.
47. Brooks B, Karplus M. Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. Proc Natl Acad Sci USA 1983;80:6571–6575.

48. Go N, Noguti T, Nishikawa T. Dynamics of a small globular protein in terms of low-frequency vibrational modes. Proc Natl Acad Sci USA 1983;80:3696–3700.
49. Levitt M, Sander C, Stern PS. Protein normal-mode dynamics: trypsin inhibitor, crambin, ribonuclease and lysozyme. J Mol Biol 1985; 181:423–447.
50. Ma J. Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. Structure 2005;13:373–380.
51. Tama F, Brooks CL. Symmetry, form, and shape: guiding principles for robustness in macromolecular machines. Annu Rev Biophys Biomol Struct 2006;35:115–133.
52. Tirion MM. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. Phys Rev Lett 1996;77:1905–1908.
53. Bahar I, Atilgan AR, Erman B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. Fold Des 1997;2:173–181.
54. Haliloglu T, Bahar I, Erman B. Gaussian dynamics of folded proteins. Phys Rev Lett 1997;79:3090–3093.
55. Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. Biophys J 2001;80:505–515.
56. Temiz A, Meirovitch N, Bahar EI. *Escherichia coli* adenylate kinase dynamics: comparison of elastic network model modes with mode-coupling ^{15}N-NMR relaxation data. Proteins 2004;57:468–480.
57. Bahar I, Radar AJ. Coarse-grained normal mode analysis in structural biology. Curr Opin Struc Biol 2005;15:586–592.
58. Tama F, Gadea FX, Marques O, Sanejouand YH. Building-block approach for determining low-frequency normal modes of macromolecules. Proteins 2000;41:1–7.
59. Tama F, Sanejouand YH. Conformational change of proteins arising from normal modes calculations. Protein Eng 2001;14:1–6.
60. Nicolay S, Sanejouand YH. Functional modes of proteins are among the most robust. Phys Rev Lett 2006;96:078104.
61. Song G, Jernigan RL. vGNM: a better model for understanding the dynamics of proteins in crystals. J Mol Biol 2007;369:880–893.
62. Yang L, Song G, Jernigan RL. Protein elastic network models and the ranges of cooperativity. Proc Natl Acad Sci USA 2009;106: 12347–12352.
63. Zheng W. A unification of the elastic network model and the gaussian network model for optimal description of protein conformational motions and fluctuations. Biophys J 2008;94:3853–3857.
64. Zhou L, Siegelbaum SA. Effects of surface water on protein dynamics studied by a novel coarse-grained normal mode approach. Biophys J 2008;94:3461–3474.
65. Lin TL, Song G. Generalized spring tensor models for protein fluctuation dynamics and conformation changes. BMC Struct Biol 2010;10:S3.
66. Kurkcuoglu O, Jernigan RL, Doruker P. Loop motions of triosephosphate isomerase observed with elastic networks. Biochemistry 2006;45:1173–1182.
67. Doruker P, Jernigan RL, Bahar I. Dynamics of large proteins through hierarchical levels of coarse-grained structures. J Comput Chem 2002;23:119–127.
68. Keskin O, Bahar I, Flatow D, Covell DG, Jernigan RL. Molecular mechanisms of chaperonin GroEL-GroES function. Biochemistry 2002;41:491–501.
69. Tama F, Miyashita O, Brooks CLI. Flexible multi-scale fitting of atomic structures into low-resolution electron density maps with elastic network normal mode analysis. J Mol Biol 2004;337:985–999.
70. Leo-Macias A, Lopez-Romero P, Lupyan D, Zerbino D, Ortiz AR. An analysis of core deformations in protein superfamilies. Biophys J 2005;88:1291–1299.
71. Wynsberghe AWV, Cui Q. Comparison of mode analyses at different resolutions applied to nucleic acid systems. Biophys J 2005;89:2939–2949.
72. Li G, Cui Q. A coarse-grained normal mode approach for macromolecules: an efficient implementation and application to ca(21)-ATPase. Biophys J 2002;83:2457–2474.
73. Zheng W, Brooks BR, Thirumalai D. Low-frequency normal modes that describe allosteric transitions in biological nanomachines are robust to sequence variations. Proc Natl Acad Sci USA 2006;103: 7664–7669.
74. Kundu S, Melton JS, Sorensen DC, Phillips GN. Dynamics of proteins in crystals: comparison of experiment with simple models. Biophys J 2002;83:723–732.
75. Ni F, Poon BK, Wang Q, Ma J. Application of normal-mode refinement to X-ray crystal structures at the lower resolution limit. Acta Crystallogr D Biol Crystallogr 2009;65:633–643.
76. Rueda M, Chacon P, Orozco M. Thorough validation of protein normal mode analysis: a comparative study with essential dynamics. Structure 2007;15:565–575.
77. Yang Q, Sharp KA. Building alternate protein structures using the elastic network model. Proteins 2009;74:682–700.
78. Micheletti C, Carloni P, Maritan A. Accurate and efficient description of protein vibrational dynamics: comparing molecular dynamics and Gaussian models. Proteins 2004;55:635–645.
79. Lu M, Poon B, Ma J. A new method for coarse-grained elastic normal-mode analysis. J Chem Theory Comput 2006;2:464–471.
80. Mendez R, Bastolla U. Torsional network model: normal modes in torsion angle space better correlate with conformation changes in proteins. Phys Rev Lett 2010;104:228103
81. Thorpe MF. Comment on elastic network models and proteins. Phys Biol 2007;4:60–63.
82. Na H, Song G. Bridging between normal mode analysis and elastic network models. Proteins 2014;82:2157–2168.
83. Feng Y, Yang L, Kloczkowski A, Jernigan RL. The energy profiles of atomic conformational transition intermediates of adenylate kinase. Proteins 2009;77:551–558.
84. Mamonova T, Hespenheide B, Straub R, Thorpe MF, Kurnikova M. Protein flexibility using constraints from molecular dynamics simulations. Phys Biol 2005;2:S137–S147.
85. Smith BT, Boyle JM, Dongarra JJ. Matrix Eigensystem routines—Eispack guide. Lecture notes in computer science, 2nd ed. Berlin: Springer; 1976.
86. Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes in C. New York: Cambridge University Press; 1992. pp 469–481.
87. Miller ST, Xavier KB, Campagna SR, Taga ME, Semmelhack MF, Bassler BL, Hughson FM. *Salmonella typhimurium* recognizes a chemically distinct form of the bacterial Quorum-sensing signal AI-2. Mol Cell 2004;15:677–687.
88. Müller CW, Schlauderer GJ, Reinstein J, Schulz GE. Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding. Structure 1996;4:147–156.
89. Berry MB, Bae E, Bilderback TR, Glaser M, Phillips JGN. Crystal structure of ADP/AMP complex of *Escherichia coli* adenylate kinase. Proteins 2006;62:555–556.
90. Sinev MA, Sineva EV, Ittah V, Haas E. Domain closure in adenylate kinase. Biochemistry 1996;35:6425–6437.
91. Schmidt MC, Samatovaa NF, Thomas K, Park B-H. A scalable, parallel algorithm for maximal clique enumeration. J Parallel Dist Mpu 2009;69:417–428.