PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

RigidFinder: A fast and sensitive method to detect rigid blocks in large macromolecular complexes

Alexej Abyzov,[1] Robert Bjornson,[2] Mihali Felipe,[1] and Mark Gerstein[1,2,3]*

[1] Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut 06520
[2] Department of Computer Science, Yale University, New Haven, Connecticut 06520
[3] Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut 06520

ABSTRACT

Advances in structure determination have made possible the analysis of large macromolecular complexes (some with nearly 10,000 residues, such as GroEL). The large-scale conformational changes associated with these complexes require new approaches. Historically, a crucial component of motion analysis has been the identification of moving rigid blocks from the comparison of different conformations. However, existing tools do not allow consistent block identification in very large structures. Here, we describe a novel method, RigidFinder, for such identification of rigid blocks from different conformations—across many scales, from large complexes to small loops. RigidFinder defines rigidity in terms of blocks, where inter-residue distances are conserved across conformations. Distance conservation, unlike the averaged values (e.g., RMSD) used by many other methods, allows for sensitive identification of motions. A further distinguishing feature of our method, is that, it is capable of finding blocks made from nonconsecutive fragments of multiple polypeptide chains. In our implementation, we utilize an efficient quasi-dynamic programming search algorithm that allows for real-time application to very large structures. RigidFinder can be used at a dedicated web server (http://rigidfinder.molmovdb.org). The server also provides links to examples at various scales such as loop closure, domain motions, partial refolding, and subunit shifts. Moreover, here we describe the detailed application of RigidFinder to four large structures: Pyruvate Phosphate Dikinase, T7 RNA polymerase, RNA polymerase II, and GroEL. The results of the method are in excellent agreement with the expert-described rigid blocks.

Proteins 2009; 00:000–000.

Key words: protein; structure; motion; rigid; block; body; method; large; complex; macromolecular.

INTRODUCTION

Advances in macromolecular crystallography and new methods of structure determination[1] have led to the high throughput determination and deposition of protein structures to the Protein Data Bank (PDB).[2] Along with the increase in the overall number of structures deposited to the PDB, the complexity of these structures also has increased, with larger macromolecular complexes consisting of dozens of subunits/polypeptides and thousands of residues deposited every month, if not every week. This data increase requires new methods for structure analysis to be capable of dealing with large individual macromolecular complexes, as well as be applicable on a large-scale to numerous structures at a reasonable time. For the means of comparative analysis, such methods also should be sensitive to capture small but biologically important differences between structures in families of similar proteins or alternative conformations of the same protein.

Analysis of molecular motions is important for studying function, including understanding mechanism of catalysis, signal transduction, and complex formation. The majority of protein motions can be classified as joined, where two or more rigid parts/blocks (domains or loops) move relative to each other. Therefore, knowledge of protein's rigid parts is a *de facto* prerequisite for analysis of protein motions that can be applied to: (i) identification of functionally important sites, (ii) description of motion trajectories, (iii) analysis of interfaces between movable parts, and, as was recently pointed out, (iv) for better motion prediction by normal mode analysis.[3] However, cases of motions not involving rigid blocks or partial structure refolding also are known.[4,5]

Methods of rigid block detection in protein structures can be divided into two broad categories: those requiring only one conformation and those requiring two or more different conformations. Although methods in the latter category can be applied to the limited subset of PDB structures, they set a gold standard for

Additional Supporting Information may be found in the online version of this article.
Grant sponsors: NIH, AL Williams Professorship funds.
*Correspondence to: Mark Gerstein, Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06520. E-mail: mark.gerstein@yale.edu
Received 13 February 2009; Revised 19 June 2009; Accepted 26 June 2009
Published online in Wiley InterScience (www.interscience.wiley.com).
DOI: 10.1002/prot.22544

the significantly more computationally extensive methods in the first category that include Molecular Dynamic simulations, Normal Mode Analysis, rigidity percolation, and various computational approaches.[6–15] In fact, if applied to predicted conformations, rigid block identification can complement analysis done by those methods. Additionally, detection of rigid blocks from two different known conformations puts motion analysis in direct biological context. Therefore, identification of movable rigid blocks from several protein conformations is important.

While several attempts to rigid block identification have been reported in the literature,[3,16–21] they are aimed at rigid block identification applicable to the movements of large domains only. The simple sieve-fit utilized in MolMovDB[19] and more complicated clustering and jump-minimizing path algorithm[16] can only detect the largest rigid block. The deformation plot analysis assumed[22] consecutive order of rigid blocks in the polypeptide chain, which is often not true. Most of the methods leave motions of small subdomains, elements of secondary structure and loops unattended because of relying on averaged values like root mean-square deviation[3,16,19,21] or Mean-Square Fluctuations.[17] When averaged over large set of residues, small structural variations, or pronounced structural variations (but only in a small subset of residues) are missed by those methods. The effect of averaging persistently increases with the size of the analyzed protein, simply because of the averaging over larger residue numbers, decreasing sensitivity of the mentioned methods and making them less applicable to detection of motions. Although, a method for sensitive rigid block analysis has been described[20] that requires approximate block size as an input or, if not given the size, performs an exhaustive search that scales exponentially with the number of residues in a protein. This makes its application to the larger macromolecular complexes practically impossible. Besides, the method produces multiple overlapping rigid blocks, thus, making interpretation of the results difficult. Another typical problem for existing methods is that they operate on a single polypeptide chain, while macromolecular complexes are essentially multi-chain. Therefore, to date there is not a sensitive and fast method that can be applied to partition large multi-chain macromolecular complexes into rigid blocks.

Here, we describe a new method, RigidFinder for identification of rigid blocks from two known conformations of a large macromolecular complex. Similar to the previous works,[18,20] we adopted the physical definition of rigidity by the distance difference between equivalent points in two conformations. As explained in the text, the definition allows for precise and sensitive identification of rigid blocks. RigidFinder is able to detect rigid blocks as small as four residues in size, and it does not have a minimal block size or number of rigid blocks as input parameters. Moreover, the novel search algorithm utilized by the method is fast enough to allow for analysis of large macromolecular complexes, and we describe the application of RigidFinder to the analysis of Pyruvate Phosphate Dikinase,[23] T7 RNA polymerase,[4,5] RNA polymerase II,[24–26] and GroEL.[27,28]

Additionally, we developed a public web server for rigid block identification by RigidFinder method (http://rigidfinder.molmovdb.org) where a user can instantly find, analyze, and visualize rigid blocks from two conformations of a protein complex. The server is interactively linked to a new multi-chain morphing server (http://morph2.molmovdb.org) with the option of using superposition by any calculated rigid block to generate a morph.

APPROACH

Definition of rigidity

In this analysis, we represent protein structures as points in the center of the residues' Cα-carbons. Given two conformations (A and B) of a protein structure, we consider a part/block consisting of N residues to be rigid if the distance difference between any two residues in the two conformations is smaller than the sensitivity cut off d

$$\left| d_{ij}^{A} - d_{ij}^{B} \right| \leq d, \quad \forall i = 1 \ldots N, \quad j = 1 \ldots N \qquad (1)$$

It is easy to see that this definition allows the development[20] of a much more sensitive method than the one that uses some kind of integral value as a definition. For instance, the characterization of rigid blocks by RMSD after the best fit of equivalent parts from two conformations is not prone to detect changes that involve a small fraction of residues, even though the absolute number of residues can be large. This is simply because those conformational changes are averaged over all fitted residues and do not significantly affect the overall RMSD, regardless of whether it is included in the fit. For the same reason rigid blocks defined by RMSD or any other integral value will tend to have regions, which are actually not rigid and can be quite different. Definition (1), on the contrary, has to be applied to every pair of residues in a block; thus, conformational change in a single one will be noticed. Also note that the definition does not assume any residue order in protein and thus can be applied to find rigid blocks consisting of nonconsecutive fragments of polypeptide chains. The downside of the definition is that n^2 distances have to be checked for each rigid block consisting of n residues. Therefore, in order to apply the definition to large macromolecular complexes an efficient algorithm for finding the optimal solution has to be developed.

Algorithm to find rigid blocks

The method operates iteratively by finding the largest rigid block in the given set of residues. All protein

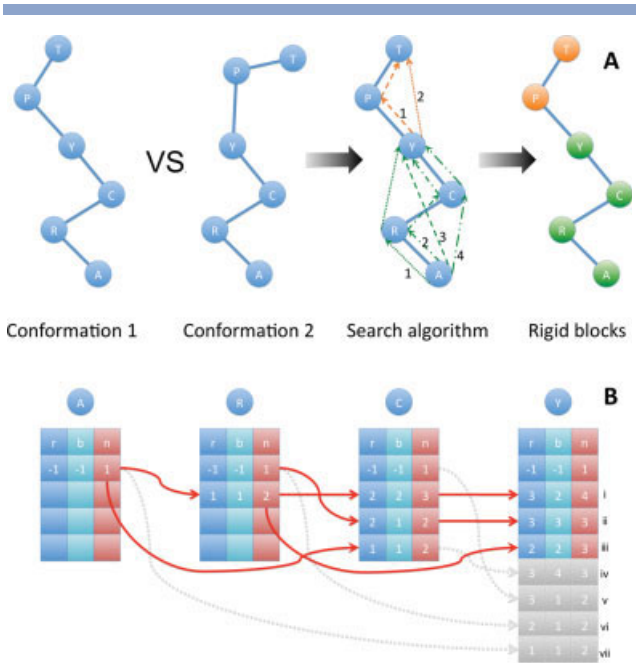

Figure 1

Schematic diagram of the RigidFinder method. (**A**) Example of all paths for block extension from residues A and Y for the comparison of two protein conformations. Paths from residue R will be redundant with paths 1 and 2 from residue A. The largest rigid block will consist of the first four residues (shown in green) and another rigid block will consist of the remaining two residues (shown in orange). (**B**) Details of quasi-dynamic programming to find the largest rigid block for the first four residues. Each block is tracked by three numbers: r – index of the previous residues in the block, b – index of the block as it is for the previous residue, and *n* – number of residues in the block. Assuming that only four blocks are tracked (the algorithm actually tracks 50), after passing residue Y four paths are not tracked (shown in gray).

residues are used to find a rigid block in the first iteration. Each of the following iterations operates on residues not assigned to rigid block(s) on previous iteration(s). The iterations stop when no rigid block of size 4 (**Methods**) or larger can be found. Every iteration proceeds in two steps: initial block detection and refinement.

At each step multiple rigid blocks are initiated at each residue and are extended, as long as rigidity condition (1) is met, by all possible paths through the residue list in the direction toward the end of the list (Fig. 1). The combinatorial complexity of block extension through all possible paths is enormous but, importantly, is redundant as paths from two residues in the same rigid block overlap. Therefore, we apply reasonable heuristics to cope with this by parallel tracking a limited number of block extensions at a time: namely, when a residue initiates a rigid block it is also considered to be added, by rigidity condition (1), to other rigid blocks initiated and extended till this point from already passed residues.

Only the largest blocks, a residue has been assigned to, are tracked [Fig. 1(B)]—thus, effectively reducing combinatorial complexity by eliminating redundant and unlikely paths (with a small number of residues) for block extension. For each such block, a number of values are stored to trace previously assigned residues in the block and to track the block size *n*. Once traversal through all the residues is completed, finding the maximum *n* and tracing back residues gives the largest block. The described algorithm is neither greedy nor a pure dynamic programming algorithm.[29] The greedy algorithm is tracking only one (best) solution at each step, that is, with index *i* on Figure 1(B). Dynamic programming would track all possible nonoverlapping solutions, that is, with indices *i* through *vii*. Our algorithm tracks a subset (more than one) of the most promising solutions, that is, with indices *i* though *iii*. We, thus, call it a "quasi-dynamic programming" algorithm.

During the first step, initial block detection, residues are put in a residues list in the same order as in a polypeptide chain from N- to C-terminal with chain order as submitted by a user. In the second step, refinement, residues are sorted in ascending order by their distance to the closest residue in the initial block (residues in the block will have zero distance and be first in the list). Refinement is necessary to avoid bias due to ordering residues by polypeptide chain. The "quasi-dynamic programming" is applied to the sorted residue list to calculate refined rigid block.

On a test data set no significant differences in block definition were observed when more than 30 and up to 100 blocks were tracked. Therefore, the algorithm conservatively tracks several more block assignments for each residue, namely the 49 largest and one initiated (a total of up to 50).

The rigid block found to be largest is postprocessed to remove and/or cluster residue fragments and gaps of length less than four residues (**Methods**). First, gaps and fragments of size 1 are removed, with gaps removed before fragments. The procedure is repeated for gaps and fragments of size 2 and 3. Finally, we require all the residues in the rigid block to be in contact with each other, that is, to form a single cluster with contact distance less than 10 Å in at least one protein conformation. The resulting rigid block is discarded if its size is less than four residues.

Rigidity condition (1) allows efficient implementation of the algorithm. All distances and distance differences can be calculated only once and saved in a matrix for use during the largest block search by quasi-dynamic programming. The method is implemented in Java language.

RESULTS

The test data set

We used a manually curated set of motions from the Database of Molecular Motions.[8] We excluded entries in the category of suspected motions and motions of nucleic acids. To demonstrate the applicability of RigidFinder to larger proteins with large conformational change, we further extended the set with three large complexes (Pyruvate

Table I
Collection of Proteins Used as Example Cases in the Text and on RigidFinder Server (http://rigidfinder.molmovdb.org)

Protein name	PDB codes	Equivalent chains	Protein size, res	Value of cutoff, Å	# of blocks	Time
Large protein and complexes						
Pyruvate phosphate dikinase	1kc7	A				
	2r82	A	872	1.75	10	~3 s
	1qln	A				
T7 RNA polymerase	1msw	D	843	2.5	8	~5 s
	1i50	ABCEFHIJKL				
RNA polymerase II	2nvq	ABCEFHIJKL	3519	2.0	15	~5 m
	1kp8	FEDCBAGHIJKLMN				
GroEL-GroES	1pcq	ABCDEFGHIJKLMN	7336	6.0	34	~1 h
	1m1y	ABCDEFGH				~35 s
Nitrogenase	2afi	ABCDEFGH	3074	2.0	8	
	1f88	AB				
Rhodopsin	3cap	BA	627	2.0	9	~3 s
Medium size proteins						
	2eck	B				
Phosphostransferase	4ake	B	214	2.5	5	~1 s
	1brd	A				
Bacteriorhodopsin	2brd	A	170	1.25	5	~1 s
	2fmq	A				
DNA polymerase beta	9ici	A	328	1.5	3	~1 s
	8adh	A				
Alcohol dehydrogenase	6adh	A	374	1.25	4	~2 s
	4mdh	A				
Malate dehydrogenase	1bmd	A	333	2.0	6	~1 s
	1dqz	A				
Antigen 85C	1dqy	A	280	1.75	3	~1 s
Aspartate aminotransferase	9aat	A				
	1ama	A	401	1.5	3	~2 s
Small proteins						
	1k9p	A				
S100A6	1k9k	A	89	1.25	7	<1 s
	5cro	A				
Cro repressor	6cro	A	61	1.25	2	<1 s
	4hvp	A				
HIV-1 protease	3hvp	A	99	1.5	2	<1 s
	1ctr	A				
Calmodulin	1cll	A	141	1.25	3	<1 s
	1idg	A				
Bungarotoxin	1idi	A	74	2.5	4	<1 s

Number of rigid blocks is given for a cutoff value corresponding to the first maximum on robustness curve for the protein. The cutoff for GroEL is chose at a global maximum. Time bench marking was done on 2.6 GHz Intel Core 2 Duo CPU at cutoff 2.5 Å.

Phosphate Dikinase, RNA polymerase II, and Nitrogenase). The resulting set consisted of 196 proteins with two different conformations (Supporting Information). Representative examples from the set for proteins of different sizes and scales of motion are listed in Table I and can be viewed at a dedicated web server (http://rigidfinder.molmovdb.org). Later in this article, we will describe the application of RigidFinder to the analysis of Pyruvate Phosphate Dikinase,[23] T7 RNA polymerase,[4,5] RNA polymerase II,[24–26] and GroEL.[27,28]

Choosing a value of sensitivity cutoff

The RigidFinder method iteratively runs a simple quasi-dynamic programming algorithm to find successively the largest blocks of residues that conform to the rigidity condition such that the difference in distance for two conformations between any two residues in a block is less than the value of sensitivity cutoff d (**Approach** section for details). The value of cutoff d defines the tolerance level between variations in structure and conformational change, thus directly affecting the method's sensitivity. In the extreme example where d is set to infinity any two conformations of any two proteins will be considered as one rigid block by the method. In the other extreme, where d is set to zero each residue (except for the case of two identical conformations) will be defined as a rigid block. The optimal value of d should be high enough for the method to ignore small structural variations but low enough for the method to notice conformational change(s) and thus detect rigid blocks. Below in the text, we will discuss the lower limit for d.

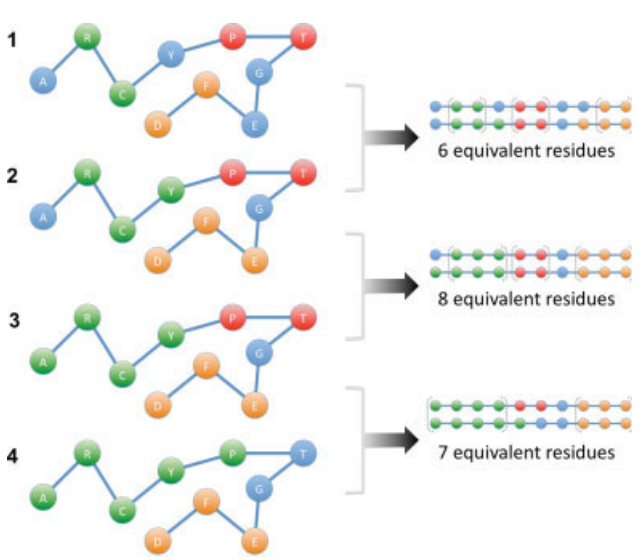

Figure 2

An example of rigid block assignments that will be reflected as local maximums on the robustness curve. Assignments 1 through 4 will correspond to a gradual increase in the value of cutoff *d*. Consistency (the number of equivalent residues) between steps 2 and 3 is the largest, since at step 4 the green block is enlarged at the expense of the red block.

It is intuitively clear that the value of *d* cannot be universally defined for all proteins, as scales of motions vary from few to 70 Å and every protein has different small variations in its structure caused by intrinsic flexibility and experimental errors. Clearly as different scales of motion can be observed in the same protein, different values of *d* can be appropriate for an analysis of the protein. To formalize this, we analyzed robustness curves calculated in the following way. We started at a cutoff value of 1 Å and gradually increased it to 6 Å with a step of 0.25 Å. At each step, we analyzed the consistency of the rigid block definition compared with the previous step—namely, we calculated and plotted the number of equivalent residues in the both definitions. The procedure to calculate equivalent residues in two block assignments is described in the **Methods** section but here we highlight its important feature that differences in both block boundaries and the number of rigid blocks are accounted for when calculating equivalent residues.

A rationale behind such an analysis is that rigid block assignments cannot be continually consistent with a steadily increasing value of *d*. Consider starting from a very sensitive block assignment (very small *d*) and decreasing sensitivity (i.e., increasing *d*). This will enlarge blocks until almost all residues will be assigned to a block. Until this "critical point," block assignments are consistent from one step to another. At the critical point a slight increase in value of *d* will not change their overall extent, that is, block partitioning has reached a saturation. However, further increase in the value of *d* will eventually reduce the method's sensitivity to such an extent that smaller blocks will need to joined into larger ones, dramatically changing block assignments. Then, with increasing values of *d*, the block assignment will remain consistent until the next critical point is reached (Fig. 3).

More specifically, consider the example shown in Figure 2. A gradual increase of cutoff for steps 1, 2, 3, and 4 leads to different block assignments. Green and orange rigid blocks are enlarged at step 2 when compared with step 1 with six equivalent residues between the assignments. Block sizes almost do not change at step 3, yielding eight equivalent residues, the maximum of all steps. At step 4, however, the green block is enlarged at the expense of the red block, with the red block not assigned anymore, leading to a decreased consistency in block assignment compared with step 3. Therefore, the best consistency between the two block assignments is archived at the local maximum number of equivalent residues. In other words, block assignment is robust to variations in sensitivity cutoff *d* in the region defined by the maximum.

Different local maximums on robustness curves [Fig. 3(A–C)] will define values of cutoff *d* for different rigid block partitioning at different sensitivity levels. While each such partitioning is of potential interest, in this work, we focused our analysis on the most sensitive partitioning (corresponding to the first local maximum). It is important that the location of the maximum defines the upper boundary of the 0.25 Å region where rigid block partitioning is similar; thus, any value in this region may be appropriate for analysis. Figure 3(D) shows the distribution of the first, second, and third local maximums on robustness curves from proteins from the test data set. Each distribution is rather broad and spans at least 2 Å. Therefore, it can be concluded that no universal cutoff can be applied for all proteins, and analysis of individual robustness curves is desirable in each case.

Quality of structures and value of sensitivity cut off

It is important to account for the fact that experimental conformations are not exact and that differences in atom pair distances are due to both a conformational change and an experimental error. As the resolution of the crystallographic data is finite, the positional uncertainties (crudely described by the B-factors) limit the precision of the atomic positions. This imposes a lower limit on the lower value of *d* at which rigid blocks can be identified meaningfully.

While there is a straightforward relationship between the value of the experimentally measured B-factor and uncertainty in the atomic coordinates, crystallographic refinements usually disregard structural heterogeneity and thus overestimate the accuracy of crystallographic

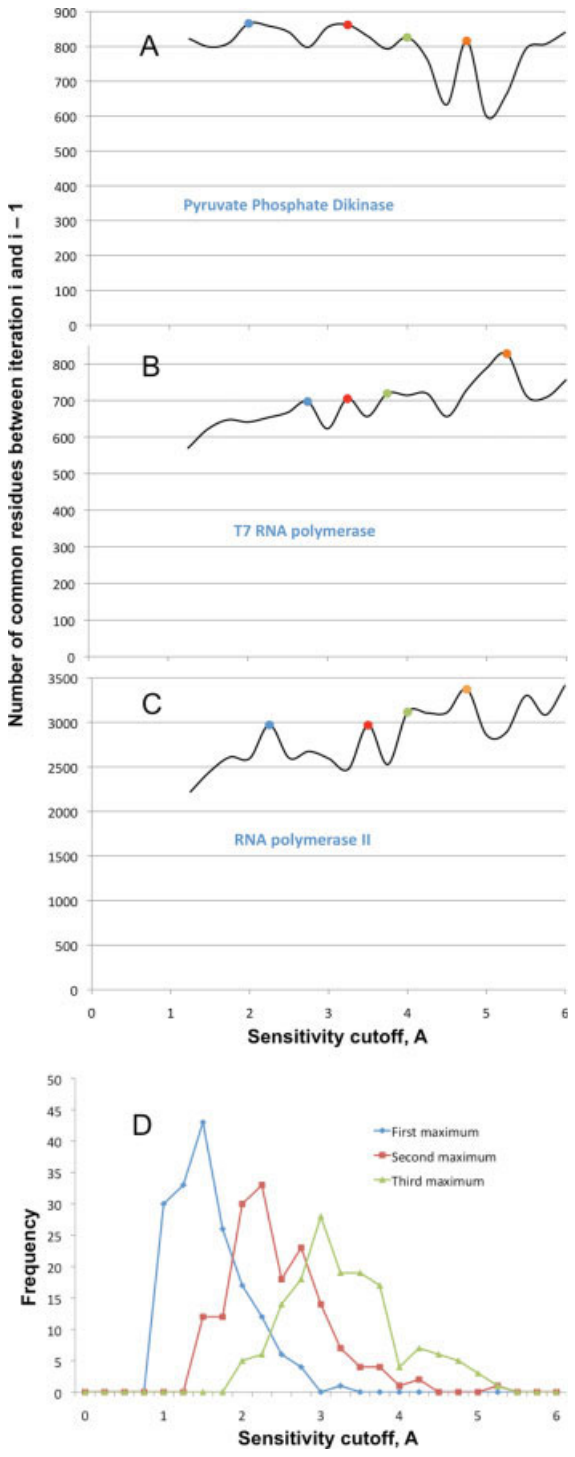

Figure 3

Robustness curves for various proteins. Critical points, i.e., local maximums, are highlighted by blue, red, green, and orange. (**A**) For Pyruvate Phosphate Dikinase. (**B**) For T7 RNA polymerase. (**C**) For RNA polymerase II. (**D**) Distribution of first, second, and third local maximum locations on robustness curves for the test data set. The quantity on ordinate for figures (A), (B), and (C) is the number of common residues between partitioning at value of cutoff *v* (termed iteration *i*) and portioning a value of cutoff *v-0.25* (termed iteration *i-1*).

structures.[30] A number of previous studies[30–32] estimated uncertainty in atomic coordinates by comparing different crystallographic structures of the same protein. While using different sets of proteins, the researchers arrived at a similar conclusion that atomic coordinates in high resolutions structures typically are precise to the extent of 0.3–0.8 Å. As RigidFinder is dealing with inter-residue distances, the lower limit for d will be, approximately, the double of those values.

To formalize this, we have calculated and plotted the distribution of residue pairwise distance differences for 10,000 random pairs of motionless protein conformations, that is for conformations that can be superimposed with RMSD of less than 2 Å (Supporting Information Figure S1). Thus, the distribution represents variations in distance difference due to experimental errors. 95% of all distance differences are less than 1 Å; therefore, to find meaningful rigid blocks, the lowest value of sensitivity cut off should be on the order of 1 Å.

Timing

For the number of residues N in a protein, a pass through the residue list by the quasi-dynamic programming algorithm will require $O(N^2)$ operations since each residue is considered for possibility to be added to a block initiated by preceding in the list residues. Each operation will be proportional to N since for each new residue added to a rigid block a rigidity condition (1) has to be checked with respect to every residue already in the rigid block, that is, all preceding residues in the worst case. Thus, the method scales as $O(N^3)$.

We empirically measured the performance of RigidFinder on the test data set. We ran the method on a single 2.6 Intel Core 2 Duo CPU. Figure 4 demonstrates that the method scales according to our estimate and that the calculation of rigid blocks for an average size protein with 100–400 residues will take less than a second to complete. Calculations for proteins of up to 1000 residues will typically complete within 10 seconds. By extrapolation one can estimate that if applied to larger proteins or complexes of up to 10,000 residues calculations will take few hours on a single CPU. This performance is likely to be improved by parallelizing the algorithm and running it on multiple CPUs.

Web server

To facilitate usage of the RigidFinder method, we set up a user-friendly web server (http://rigidfinder.molmovdb.org) for rigid block analysis using the RigidFinder method (Fig. 5). A user can upload structures for analysis or simply provide PDB-code for structures from PDB.[2] The server will then perform an input check and, if successful, generate a web page for on the-fly interactive rigid block visualization, calculation, and analysis. The

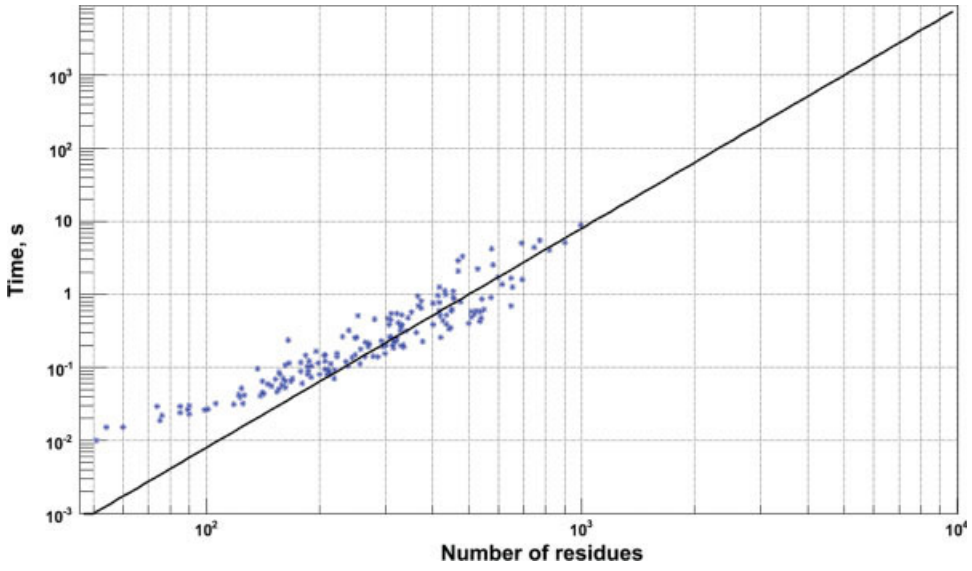

Figure 4

Measurement of RigidFinder performance. Each star on the plot represents a performance measurement for a protein from the test data set. The black solid line displays the linear regression by the $O(N^3)$ line, i.e., analytically predicted performance. The regression was done for points that correspond to proteins of at least 200 residues long. By extrapolation one can estimate that if applied to larger proteins or complexes of up to 10,000 residues calculations will take few hours on a single CPU. The benchmarking was performed on a 2.6 Intel Core 2 Duo CPU.

interactive nature of the page is especially important as the user has the opportunity to adjust the method's parameters to get the best results.

Once calculated, rigid blocks can be instantly highlighted by applying different colors or line thicknesses. Simple but effective 3D structure visualization allows instant view manipulation with a mouse. As an additional control feature, each block can be fitted to minimize RMSD between corresponding residues. The web page also has several buttons with associated scripts for quick view manipulation. Results of a rigid block assignment can be printed as residue fragments or as a PDB-file with block numbers in the occupancy field.

Visualization and the RigidFinder method are implemented in the Belka applet embedded into the page, which allows performing analyses or saving the whole web page for further analysis on the client machine, independent of server and network. Coupled with the efficient implementation of the underlying algorithm (typical rigid block calculation is done within seconds on a fairly average laptop) the generated page provides a fast, easy and convenient way for analysis.

The RigidFinder server is also interactively linked to a new multi-chain morphing server (http://morph2.molmovdb.org). A user can submit structures for morphing using current superimposition (global or for a particular rigid block). The morphing server will then produce and visualize a morphing trajectory by linear interpolation between submitted structures.

The standalone application for visualization with the integrated RigidFinder method is freely available and can be downloaded from the same web server.

Examples

To demonstrate the abilities of RigidFinder, we give here several examples of its application to four large-scale motions in macromolecules, namely the analysis of Pyruvate Phosphate Dikinase,[23] T7 RNA polymerase,[4,5] RNA polymerase II,[24–26] and GroEL-GroES.[27,28] For every protein/complex, we chose the value of sensitivity cut-off d at the first local maximum on the robustness curve. We then compared the partitioning of protein/complexes into rigid blocks with the annotation described in the literature and with partitioning produced by DynDom[17] and Hingefind.[21] Information for every example is summarized in Table I. Each example can be viewed interactively at http://rigidfinder.molmovdb.org.

Pyruvate phosphate dikinase

Pyruvate Phosphate Dikinase (PPDK) catalyzes the reversible conversion of phosphoenolpyruvate (PEP), AMP, and P^i to pyruvate and ATP. The enzyme consists of four domains and contains two remotely located reaction centers: the nucleotide partial reaction takes place at the N-terminal domain, and the PEP/pyruvate partial reaction takes place at the C-terminal domain. A His-domain tethered to the N- and C-terminal domains by two

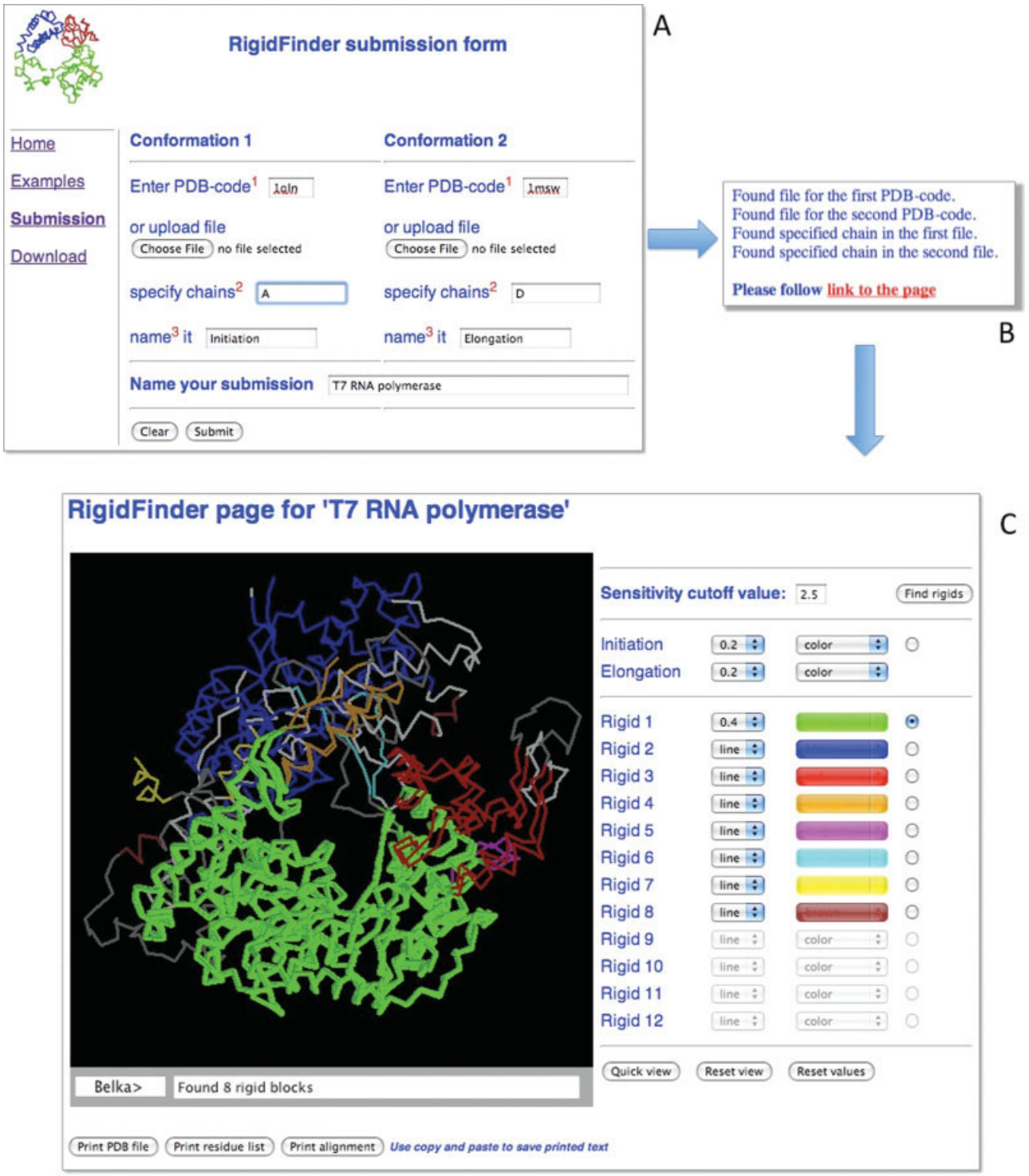

Figure 5

Web server for rigid block identification using the RigidFinder method. (**A**) A user can upload structures for analysis or simply provide PDB-code for structures from PDB. (**B**) The server performs an input check and, if successful, generates a web page for on the-fly interactive rigid block analysis. (**C**) The generated page.

closely associated linkers contains a phosphorylatable histidine residue. The central domain swivels to shuttle a phosphoryl group between the two reaction centers.[23]

By locating the first local maximum on the robustness curve [Fig. 3(A)], we set 1.75 Å as the most suitable value of sensitivity cutoff d to calculate rigid block parti-

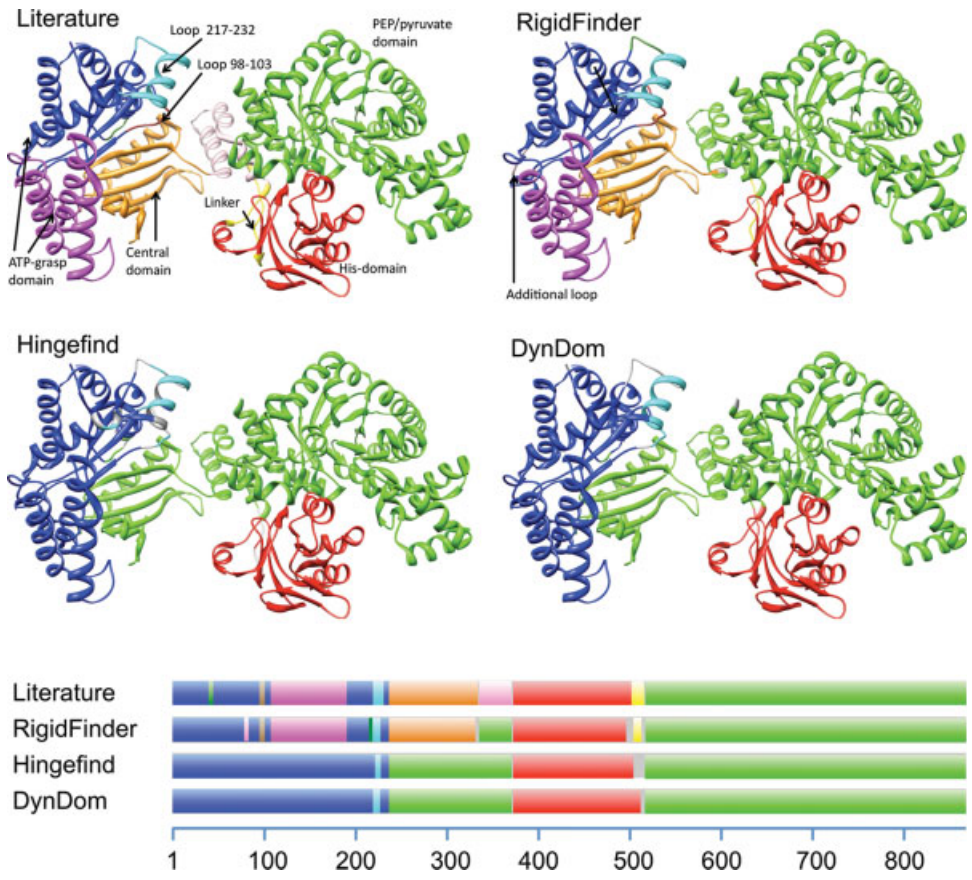

Figure 6

Comparison of rigid block assignment in pyruvate phosphate dikinase made by RigidFinder, Hingefind, and DynDom with literature-annotated domains and movable parts. Rigid blocks are highlighted by different colors. Linear diagram at the bottom shows projection of rigid blocks on polypeptide chains. RigidFinder found 10 rigid blocks, with five corresponding to the four domains of PPDK: PEP/pyruvate domain (in green), His-domain (in red), ATP-grasp domain with two movable subdomains (in blue and magenta), and central domain (in orange); four corresponding to loops that change orientation (in cyan, brown, pink, and dark green); and one corresponding to the tethering linker of His-domain (in yellow). Results are in very good agreement with the description provided in.[23] Hingefind and DynDom provide less detail on the possibly functional relevant small blocks.

tioning for PPDK. At this value, comparison of the two protein conformations revealed 10 rigid blocks (Fig. 6). Five of the blocks corresponded to the protein domains and two subdomains of the fourth domain, four corresponded to three loops that change orientation, and one corresponded to the tethering linker of His-domain. Two loops were previously reported as having conformation change. In the region of the first one (residues 217–232), RigidFinder found two rigid blocks (residues 218–228 and 229–232), thus indicating that the motion of the loop is the superposition of the two motions. The second loop (residues 98–103) corresponded exactly to one rigid block. Additionally, RigidFinder reported another rigid block (residues 81–85) that represents a loop located on the outer surface of the ATP-grasp domain, far away from the active sites.

The fact that RigidFinder was able to find all the functionally important elements that were previously described[23] without prior knowledge of the number and sizes of those elements is remarkable. DynDom and Hingefind were unable to reach such a fine level of partitioning; with each method producing only four rigid blocks and completely missing the motion of central domain, the linker, loop 98–103 and subdomains in the ATP-grasp domain.

T7 RNA polymerase

T7 RNA polymerase is an 883-residues-long protein capable of initiating and elongating RNA transcripts. Initiation and elongation are accomplished by two different protein conformations, and transition from one to another is accompanied by a massive motion and refolding of 300 N-terminal residues.[4,5]

By locating the first local maximum on the robustness curve [Fig. 3(B)], we set 2.5 Å as the most suitable value

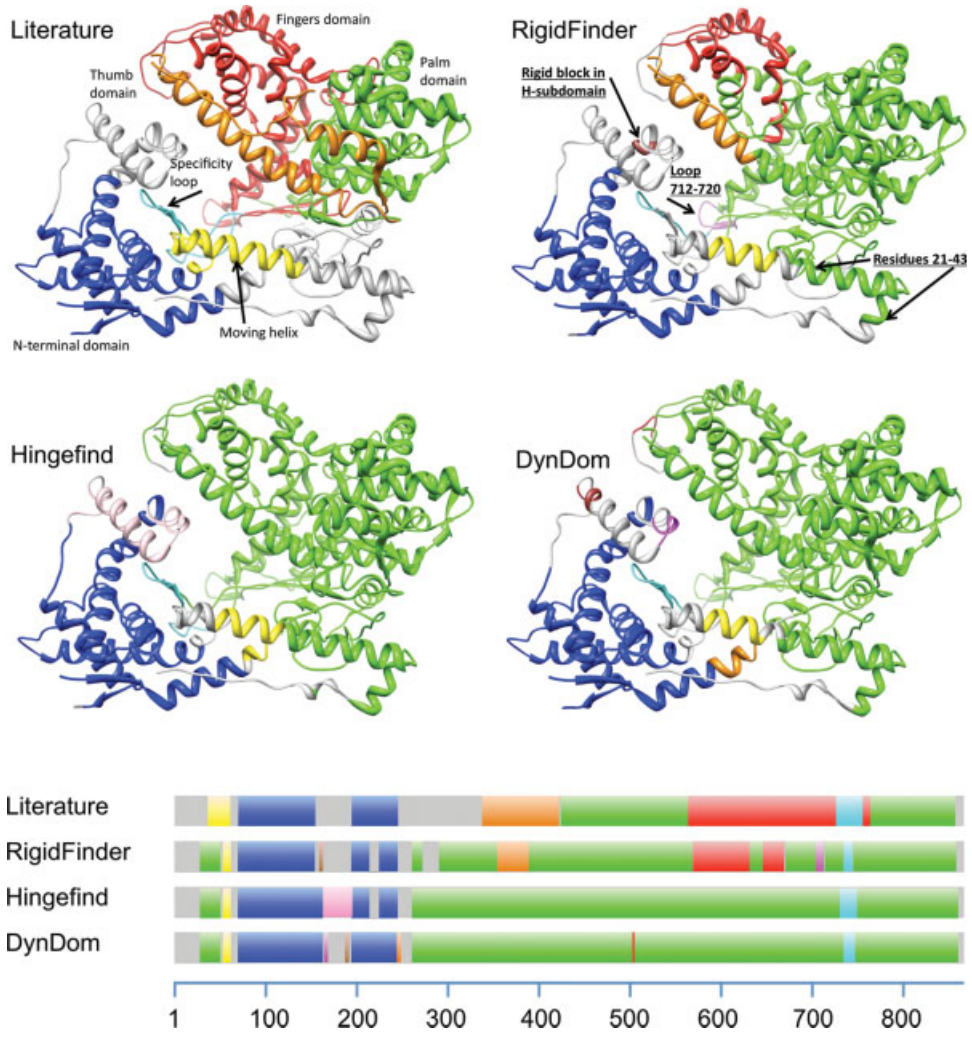

Figure 7

Comparison of rigid block assignment in T7 RNA polymerase made by RigidFinder, Hingefind, DynDom with literature-annotated movable parts. Rigid blocks are highlighted by different colors. Linear diagram at the bottom shows projection of rigid blocks on polypeptide chains. RigidFinder found eight rigid blocks, with four corresponding to the four domains: "palm" domain (in green), "fingers" domain (in red), "thumb" domain (in orange), and N-terminal domain (in blue); and four corresponding to loops and refoldable regions (in yellow, magenta, cyan, and brown). RigidFinder demonstrates the best agreement with the descriptions provided in Refs. 4 and 5.

of sensitivity cutoff d to calculate rigid block partitioning for this protein. A comparison of the two protein conformations revealed eight rigid blocks (Fig. 7). The largest rigid block corresponded to the "palm" domain, while a slight closing of the "thumb" and "fingers" domains resulted in detection of a rigid block for each (this motion was not detected by either DynDom and Hingefind). Additionally, RigidFinder reported a block for loop 712–720, which changes its conformation as a rigid body. The specificity loop (residues 739–772) largely refolds but residues 743–752 have the same conformation and are reported as a rigid block. The large motion of an N-terminal domain results in its easy detection as a second largest rigid block.

Another region (residues 7–71) undergoes a complex transition. The position and conformation of residues 21–43 does not change, and RigidFinder includes them in the largest rigid block along with the "palm" domain. The region of residues 47–56 also does not change conformation but flips by about 90° to form a single helix with residues 21–43. Considering that residues in these two stretches are in direct contact with the largest moving domain of the polymerase they may play important role in the modulation of conformational transition. Other residues in region 7–71 are not included in any rigid block.

Finally, only RigidFinder was able to detect a rigid block in the H-subdomain (residues 151–190), which was previous believed to completely refold. While the block is

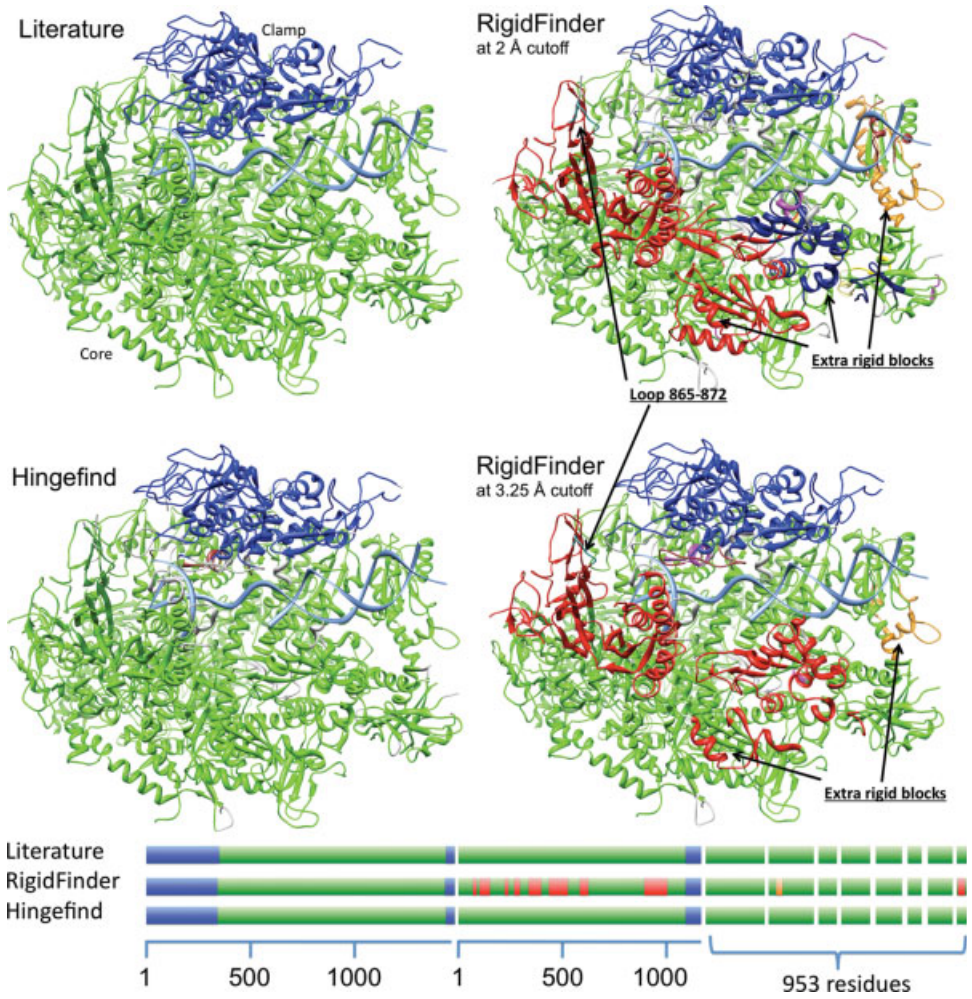

Figure 8

Comparison of rigid blocks assignment in RNA Polymerase II made by RigidFinder, Hingefind, and literature annotation. Linear diagram at the bottom shows projection of rigid blocks on polypeptide chains. Lengths of short chains are not proportional. Two different RigidFinder partitions, for cutoff 2 and 3.25 Å, are shown. There are 15 and 10 blocks, respectively. Rigid blocks are highlighted by different colors with all small blocks colored in black. RigidFinder found previously uncharacterized motions of regions wrapping the DNA helix.

small (residues 159–163), it may be essential by serving as a folding core while the folding itself can be modulated either by the length of the initiated RNA transcript or movement of the C-terminal domain.

In conclusion, the RigidFinder method detected all known and one extra feature of structural rearrangements up to the resolution of loops in T7 RNA polymerase. As in the previous example, the result was achieved without prior knowledge of the number and sizes of those elements.

RNA polymerase II

The complex of RNA polymerase II consists of 12 polypeptide chains with lengths ranging from 46 to 1423 residues, with a total of 3627 residues refined in a crystal structure. Free and elongation structures are mainly different in the orientation of clamp.[33]

We used the location of the first maximum on the robustness curve [Fig. 3(C)] to set 2 Å as the most suitable value of sensitivity cutoff d to calculate rigid block partitioning for this protein. We also considered partitioning at the second maximum, that is, at a 3.25 Å value of cutoff, to confirm residue shift around the DNA helix (see below).

With a value 2 Å of sensitivity cutoff, RigidFinder found 15 rigid blocks (Fig. 8). It clearly identified that the majority of the complex (~2200 residues) holds as a rigid body and clamp swing is the second rigid body. Several other rigid blocks found included regions around the DNA helix, a total of ~600 residues that shift by a few angstroms in elongation conformation toward the helix to wrap it tighter for, presumably, better processivity. Remarkably, RigidFinder was sensitive enough to detect this shift that has not been noticed before. The shift was not annotated in the literature nor detected by

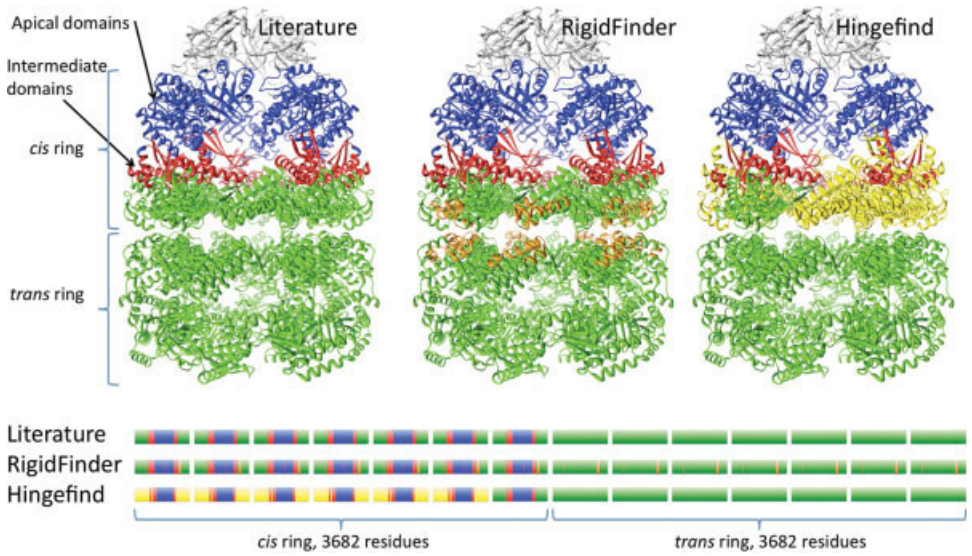

Figure 9

Comparison of rigid blocks assignment for GroEL-GroES chaperonin made by RigidFinder, Hingefind, and literature annotation. Linear diagram at the bottom shows projection of rigid blocks on polypeptide chains. Each chain is 526 residues long. Blocks found by RigidFinder are in excellent agreement with known annotation,[28,34] but RigidFinder identifies additional rigid blocks (shown in orange). Hingefind is not able to find all residues in the largest rigid block (shown in green) that includes the *trans* ring and equatorial domains from the *cis* ring. Hingefind misses six equatorial domains and reports them as separate rigid blocks (shown in yellow). Boundaries of these blocks are also imprecise (linear diagram on the bottom). Similarly, boundaries of apical (shown in blue) and intermediate (shown in red) domains are not correct.

Hingefind (DynDom cannot handle multiple chain proteins). However, we argue that this motion is real as RigidFinder finds rigid blocks, in the same location even at decreased sensitivity (cutoff value of 3.25 Å).

The remaining rigid blocks represent various loops that only slightly alter their conformation in two structures. Most of them become parts of larger rigid blocks when sensitivity is reduced, which we attribute to the low resolution of RNA polymerase II crystal structures (resolution ~2.8 Å with R-free of ~0.28). However, loop 865–872 was consistently detected at both cutoffs. Therefore, RigidFinder demonstrated excellent agreement with published results and was able to detect novel motion of large regions adjacent to the DNA helix and alterations in loop conformations.

GroEL/GroEL-GroES

GroEL and GroEL-GroES are the two conformations of *E.coli*'s chaperonin complex, assisting protein folding with the consumption of ATP. The GroEL complex consists of 14 identical chains of 526 residues each, comprising two symmetrical rings (*cis* and *trans*) stacked back to back. Co-chaperonin GroES binds to the *cis* ring and stabilizes the massive movement of domains in the ring to form a folding chamber.[27,28]

We identified rigid blocks in GroEL/GroEL-GroES by comparing both conformations of individual chains (at 2.5 Å of cutoff) and the whole complexes. When comparing individual chains, RigidFinder clearly identified three rigid blocks corresponding to the known domains of GroEL chains. For analysis of the whole complexes, we used 6 Å as cutoff. We chose this value because we observed that the most robust partitioning is observed when the value of sensitivity cutoff is between 5.5 and 7 Å (Supporting Information Figure S2).

A comparison of GroEL and GroEL-GroES complexes revealed 34 rigid blocks (Fig. 9). The largest block included the *trans* ring and all equatorial domains from chains comprising the *cis* ring (one for each chain). Fourteen more blocks corresponded to the intermediate and apical domains of the *cis* ring (two for each chain). Seven other blocks included residues from regions 420–445 and sometimes 105–120 from equatorial domains of the *cis* ring. Surprisingly, the same regions in the *trans* ring were detected in only four chains and were by 5–10 residues shorter, which may be due to a slight asymmetry of structures. Remarkably, those regions represent conformational changes not characterized before. In both rings, the described rigid blocks are located on the ring interface and involved in cross-ring interaction. Other rigid blocks corresponded to smaller loops with sporadic locations in the structure. Because of the decreased method sensitivity and nonsymmetrical location of those loops, we question their relevance to the analysis and do not describe them here.

Hingefind produced partitioning similar to RigidFinder and literature description. However, the largest rigid block (consisting of *trans* ring and all equatorial

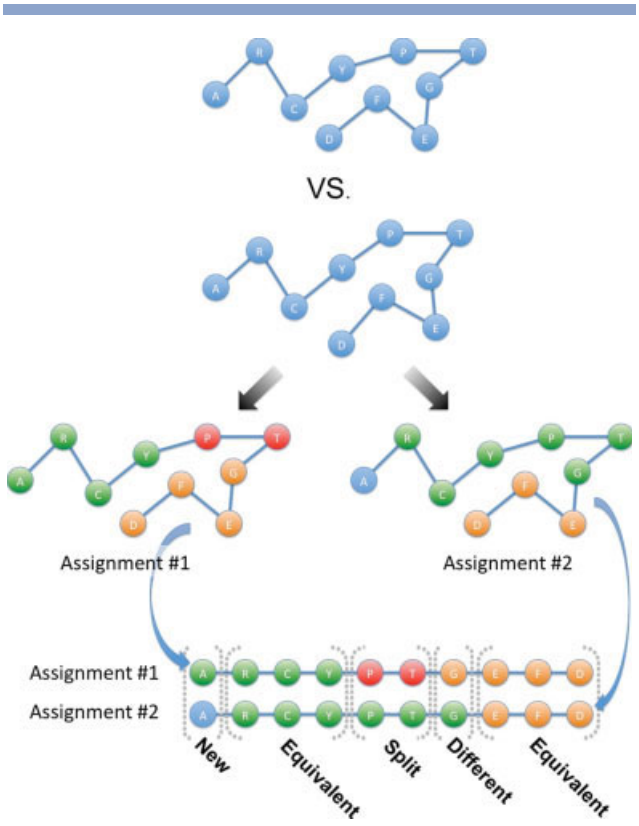

Figure 10

Explanation of residues categories for method comparison. This hypothetical example shows two assignments (Assignment #1 and Assignment #2) with three and two rigid blocks, respectively. Blocks are colored green, red, and orange. Residues in overlapping assignments can be classified as equivalent, split, or different. If both blocks have largest mutual overlap then overlapping residues are classified as equivalent, as in case of green and orange blocks. Red block by Assignment #1 has largest overlap with the green block by Assignment #2 but the opposite is not true. Such overlapping residues are classified as split. All other overlapping residues are classified as different, i.e., those that are not in equivalent or split categories. New residues are those that are assigned to a rigid block by only one method.

domains) was split into two highly nonsymmetrical rigid blocks, namely: one consisting of *trans* ring and one equatorial domain and the other one from six equatorial domains. Taking into account that GroEL and GroEL-GroES complexes are symmetric, we feel that such partitioning is not reasonable.

DISCUSSION

In this work, we developed and applied a novel method, RigidFinder, for the assignment of movable rigid blocks from two different conformations of large macromolecular, multiple, and single chain proteins. With the aim of developing a sensitive method, we considered a block to be rigid if all inter-residue distances are almost the same, that is, the difference in distance between any two residues in a block is less than the value of sensitivity cutoff *d*. Following this definition we applied an efficient heuristic algorithm to find the largest rigid block. Iterative application of the algorithm identifies multiple rigid blocks in a given protein structure.

The method has been tested on proteins from the Database of Molecule Motions and its application to several macromolecular proteins, Pyruvate Phosphate Dikinase, T7 RNA Polymerase, RNA polymerase II, and GroEL have been described. In all cases the results of RigidFinder were in excellent agreement with previously described motions and function annotation for each of the proteins. However, RigidFinder was able to detect previously unreported rigid blocks that can potentially have functional implications. It was demonstrated that RigidFinder is indeed very sensitive and, for two given conformations of a protein, it is capable of detecting rigid blocks in the full range of sizes from large domains to loops as small as four residues long. The minimal size of four residues arises as a natural limitation due to the inherent variability in protein structure. Remarkably, the method does not require a specific number of rigid blocks as an input; thus, the search for rigid blocks by RigidFinder is exhaustive and objective.

Comparison of RigidFinder results against results from DynDom[17] and Hingefind,[21] other methods aimed at movable rigid block identification, revealed that RigidFinder provides substantially more details on possible functional small blocks. This result is not surprising as both DynDom and Hingefind were designed to identify large domains/blocks that exhibit clearly identifiable rigid-body movements and are likely to ignore small structural variations.

The analysis of rigid blocks by RigidFinder has a few limitations. Experimental error in atom positioning imposes the lower limit on sensitivity cut off, which, we estimated, is around 1 Å. Also, the fact that the method is more sensitive to smaller rigid blocks obviously increases the risk that experimental artifacts affect it.

To facilitate usage of the RigidFinder method, we set up a user-friendly web server for rigid block analysis using RigidFinder at (http://rigidfinder.molmovdb.org; Fig. 5). Efficient implementation of the underlying search algorithm allows on the-fly identification and visualization of rigid blocks, thus allowing a user to quickly sample values of sensitivity cutoff *d* to find the optimal one. An additional benefit of the server is that a user does not need to install any software and can work from his/her favorite browser or save the whole web page for local use. The server is interactively linked to the new multi-chain morphing server (http://morph2.molmovdb.org) with the option of using superposition by any calculated rigid block to generate a morph.

Besides applying the RigidFinder method to the analysis of motions in individual macromolecular protein complexes, it can be used in the large-scale analysis of motions and to improve motions predictions by NMA

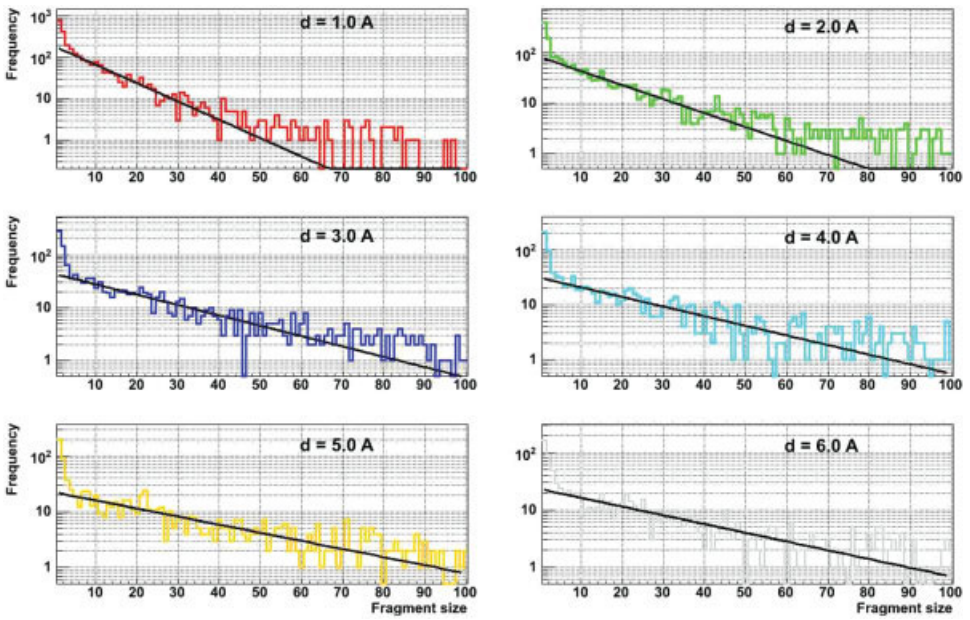

Figure 11

Distributions of fragment sizes comprising the largest rigid block in each protein of the test data set at different values of sensitivity cutoff *d*. The solid black line represents the fit of distributions by exponents. The distributions are very different from exponents in regions 1–3.

analysis.[3] Information about rigid blocks can also be used for better morphing between protein conformations. Thus, RigidFinder is a novel, unique, convenient, and freely available tool for the scientific community.

METHODS

Methodology to compare different block assignments

Given two rigid block assignments (termed Assignment #1 and Assignment #2) we introduced four different categories of residues for comparison (Fig. 10):

- Equivalent—overlapping residues from blocks of Assignment #1 and Assignment #2 that have the largest mutual overlap;
- Split—overlapping residues from blocks of Assignment #1 and Assignment #2 where only one block has the largest overlap with the other;
- Different—overlapping residues from blocks of Assignment #1 and Assignment #2 that cannot be classified as equivalent or split;
- New—residues not assigned to any rigid block by one of the methods.

The suggested set of categories represents a comprehensive classification of variations between two given block assignments with a direct quantification of differences and similarities. If most of the residues are equivalent then the two assignments are similar. When assignments are different then the nature of differences can be revealed: split residues will indicate split(s) in rigid blocks into two or more smaller blocks, while different residues will indicate different block boundaries.

Defining the size of short fragments and gaps to be removed/clustered during post-processing

Rigid blocks do not necessarily consist of a single polypeptide fragment. Consequently, a rigid block can be described as a set of fragments of the polypeptide chain with gaps between. However, flexibility and random errors occurring during structure determination lead to natural variations in protein structure. Because of these variations, residues with inter-residue distances near the threshold can potentially be included or excluded "by chance." This increases the fragmentation of rigid blocks. It is intuitively clear that these gaps will be short. To study this effect and determine the optimum "gap/fragment size," we analyzed the fragmentation of rigid blocks on the test data set.

The RigidFinder algorithm was applied to every pair of conformations in the test data set with different values of sensitivity cutoff *d*. Distributions of fragment and gap sizes for the largest rigid block were then produced (Figs. 11 and 12). All the distributions can be well described by exponents except for region 1–3, where the frequency of events is almost a full order of magnitude higher than expected by interpolating exponents. These fragments and gaps of lengths from 1 to 3 represent the natural variation in the protein structure, while larger

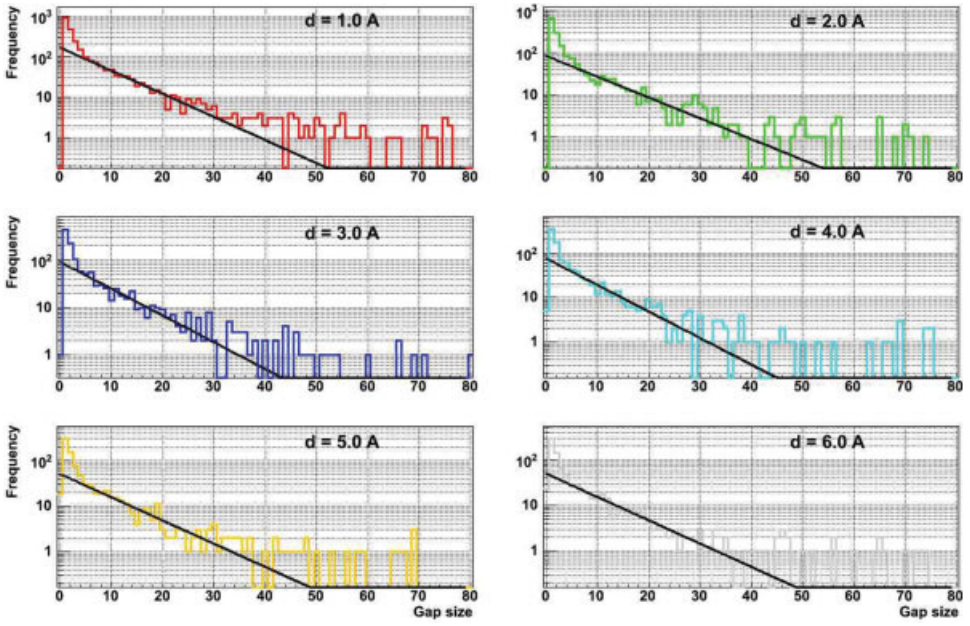

Figure 12

Distributions of gap sizes between fragments comprising the largest rigid block in each protein of the test data set at different values of sensitivity cutoff *d*. The solid black line represents the fit of distributions by exponents. The distributions are very different from exponents in regions 1–3.

fragments, with distribution described by exponents, represent fragmentation due to the chosen cutoff *d*. On the basis of this analysis, we clustered/removed from rigid block assignment fragments and gaps of sizes up to three, as those are likely to represent variations and random residues in protein structure rather than motions.

Figures preparation

Pictures were prepared with the aid of the ROOT data analysis framework (http://root.cern.ch), an extensive molecular modeling package by Chimera,[35] and the freely available program Belka that has RigidFinder integrated (http://rigidfinder.molmovdb.org).

REFERENCES

1. Dutta S, Berman HM. Large macromolecular complexes in the protein data bank: a status report. Structure 2005;13:381–388.
2. 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.
3. 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.
4. Tahirov TH, Temiakov D, Anikin M, Patlan V, Mcallister WT, Vassylyev DG, Yokoyama S. Structure of a T7 RNA polymerase elongation complex at 2.9 A resolution. Nature 2002;420:43–50.
5. Yin YW, Steitz TA. Structural basis for the transition from initiation to elongation transcription in T7 RNA polymerase. Science 2002; 298:1387–1395.
6. 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.
7. Dumontier M, Yao R, Feldman HJ, Hogue CW. Armadillo: domain boundary prediction by amino acid composition. J Mol Biol 2005; 350:1061–1073.
8. 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.
9. Flores SC, Gerstein MB. FlexOracle: predicting flexible hinges by identification of stable domains. BMC Bioinformatics 2007;8:215.
10. 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.
11. Flores SC, Lu LJ, Yang J, Carriero N, Gerstein MB. Hinge Atlas: relating protein sequence to sites of structural flexibility. BMC Bioinformatics 2007;8:167.
12. Jacobs DJ, Rader AJ, Kuhn LA, Thorpe MF. Protein flexibility predictions using graph theory. Proteins 2001;44:150–165.
13. Kundu S, Sorensen DC, Phillips GN. Automatic domain decomposition of proteins by a Gaussian network model. Proteins 2004;57: 725–733.
14. Painter J, Merritt EA. A molecular viewer for the analysis of TLS rigid-body motion in macromolecules. Acta Crystallogr D 2005; 61:465–471.
15. Shatsky M, Nussinov R, Wolfson HJ. Flexible protein alignment and hinge detection. Proteins 2002;48:242–256.
16. Boutonnet NS, Rooman MJ, Wodak SJ. Automatic analysis of protein conformational changes by multiple linkage clustering. J Mol Biol 1995;253:633–647.
17. 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.

18. Hinsen K, Thomas A, Field MJ. Analysis of domain motions in large proteins. Proteins 1999;34:369–382.
19. Krebs WG, Gerstein M. The morph server: a standardized system for analyzing and visualizing macromolecular motions in a database framework. Nucleic Acids Res 2000;28:1665–1675.
20. Nichols WL, Rose GD, Ten Eyck LF, Zimm BH. Rigid domains in proteins: an algorithmic approach to their identification. Proteins 1995;23:38–48.
21. 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.
22. Huang ES, Rock EP, Subbiah S. Automatic and accurate method for analysis of proteins that undergo hinge-mediated domain and loop movements. Curr Biol 1993;3:740–748.
23. Lim K, Read RJ, Chen CC, Tempczyk A, Wei M, Ye D, Wu C, Dunaway-Mariano D, Herzberg O. Swiveling domain mechanism in pyruvate phosphate dikinase. Biochemistry 2007;46:14845–14853.
24. Cramer P, Bushnell DA, Fu J, Gnatt AL, Maier-Davis B, Thompson NE, Burgess RR, Edwards AM, David PR, Kornberg RD. Architecture of RNA polymerase II and implications for the transcription mechanism. Science 2000;288:640–649.
25. Cramer P, Bushnell DA, Kornberg RD. Structural basis of transcription: RNA polymerase II at 2.8 angstrom resolution. Science 2001; 292:1863–1876.
26. Wang D, Bushnell DA, Westover KD, Kaplan CD, Kornberg RD. Structural basis of transcription: role of the trigger loop in substrate specificity and catalysis. Cell 2006;127:941–954.
27. Braig K, Adams PD, Brunger AT. Conformational variability in the refined structure of the chaperonin GroEL at 2.8 A resolution. Nat Struct Biol 1995;2:1083–1094.
28. Xu Z, Horwich AL, Sigler PB. The crystal structure of the asymmetric GroEL-GroES-(ADP)7 chaperonin complex. Nature 1997;388: 741–750.
29. Bellman R, Kalaba R. Dynamic programming and statistical communication theory. Proc Natl Acad Sci USA 1957;43:749–751.
30. Depristo MA, De Bakker PI, Blundell TL. Heterogeneity and inaccuracy in protein structures solved by X-ray crystallography. Structure 2004;12:831–838.
31. Eyal E, Gerzon S, Potapov V, Edelman M, Sobolev V. The limit of accuracy of protein modeling: influence of crystal packing on protein structure. J Mol Biol 2005;351:431–442.
32. Flores TP, Orengo CA, Moss DS, Thornton JM. Comparison of conformational characteristics in structurally similar protein pairs. Protein Sci 1993;2:1811–1826.
33. Gnatt AL, Cramer P, Fu J, Bushnell DA, Kornberg RD. Structural basis of transcription: an RNA polymerase II elongation complex at 3.3 A resolution. Science 2001;292:1876–1882.
34. Braig K, Otwinowski Z, Hegde R, Boisvert DC, Joachimiak A, Horwich AL, Sigler PB. The crystal structure of the bacterial chaperonin GroEL at 2.8 A. Nature 1994;371:578–586.
35. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE. UCSF Chimera—a visualization system for exploratory research and analysis. J Comput Chem 2004;25:1605–1612.