PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

HingeMaster: Normal mode hinge prediction approach and integration of complementary predictors

Samuel C. Flores,[1,2*] Kevin S. Keating,[3] Jay Painter,[4] Faruck Morcos,[5] Khang Nguyen,[2] Ethan A. Merritt,[4] Leslie A. Kuhn,[6,7,8] and Mark B. Gerstein[2,3,9*]

[1] Department of Physics, Yale University, Bass 432, New Haven, Connecticut 06520
[2] Department of Molecular Biophysics and Biochemistry, Yale University, Bass 432, New Haven, Connecticut 06520
[3] Program in Computational Biology and Bioinformatics, Yale University, Bass 432, New Haven, Connecticut 06520
[4] Department of Biochemistry, University of Washington, Mailstop 357742, Seattle, Washington 98195-7742
[5] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, Indiana 46556
[6] Department of Biochemistry & Molecular Biology, Michigan State University, East Lansing, Michigan 48824
[7] Department of Computer Science & Engineering, Michigan State University, East Lansing, Michigan 48824
[8] Department of Physics & Astronomy, Michigan State University, East Lansing, Michigan 48824
[9] Department of Computer Science, Yale University, Bass 432, New Haven, Connecticut 06520

ABSTRACT

Protein motion is often the link between structure and function and a substantial fraction of proteins move through a domain hinge bending mechanism. Predicting the location of the hinge from a single structure is thus a logical first step towards predicting motion. Here, we describe ways to predict the hinge location by grouping residues with correlated normal-mode motions. We benchmarked our normal-mode based predictor against a gold standard set of carefully annotated hinge locations taken from the Database of Macromolecular Motions. We then compared it with three existing structure-based hinge predictors (TLSMD, StoneHinge, and FlexOracle), plus HingeSeq, a sequence-based hinge predictor. Each of these methods predicts hinges using very different sources of information—normal modes, experimental thermal factors, bond constraint networks, energetics, and sequence, respectively. Thus it is logical that using these algorithms together would improve predictions. We integrated all the methods into a combined predictor using a weighted voting scheme. Finally, we encapsulated all our results in a web tool which can be used to run all the predictors on submitted proteins and visualize the results.

Proteins 2008; 73:299–319.

Key words: protein; motion; domain; hinge; bending; atlas; flexibility; conformation; change; molmovdb.

INTRODUCTION

The structural information provided by the atomic coordinates of a protein tells only part of the story of protein function. Much of the remainder is told by the trajectory of motion. Motions can be classified according to the size of the mobile units, which may be fragments, domains, or subunits, and according to packing as hinge, shear, or other. Hinge bending motions are the largest single class of motions, comprising 45% of the motions in a representative set.[1,2] This class is further subdivided into domain hinge motions (31% of the total)[1] and fragment hinge motions (13%). A logical first step towards the goal of motion prediction for the case of domain hinge bending motions is to predict the hinge location. In this work, we compare several existing algorithms, present new ones, and combine all of these into HingeMaster, a composite predictor.

The problem of hinge detection is easiest when two or more sets of atomic coordinates are available for a given protein in different conformations. In that case, it is possible to visually inspect the pairs of structures (as we have done in this work) and manually annotate the hinge location. It is also possible to automate this using FlexProt or (to some extent) other algorithms.[3] Hinge detection based on two structures is thus largely a solved problem. A much more challenging problem arises when only *one* structure is known. In early work on this problem, Janin and Wodak[4] developed a domain interface area

Additional Supporting Information may be found in the online version of this article.
Grant sponsor: National Institutes of Health (NIH); Grant sponsor: Simbios, the NIH Roadmap for Medical Research; Grant number: U54 GM072970; Grant sponsor: National Library of Medicine; Grant number: T15 LM07056.
*Correspondence to: Samuel C. Flores, 266 Whitney Ave., New Haven, CT 06520. E-mail: samuel.flores@aya.yale.edu; or Mark B. Gerstein, Yale University, Bass 432, New Haven, CT 06520. E-mail: mark.gerstein@yale.edu
Received 1 June 2007; Revised 27 December 2007; Accepted 21 February 2008
Published online 23 April 2008 in Wiley InterScience (www.interscience.wiley.com).
DOI: 10.1002/prot.22060

calculation method. The more recent FIRST algorithm[5–9] identifies rigid substructures based on graph theoretic calculations. FRODA uses these rigid units to simplify the process of generating alternate structures which have been shown to be consistent with NMR data for certain proteins.[10,11] The Gaussian Network Model (GNM)[12] is an approximate method for obtaining normal mode displacements and consequent motional correlations of backbone α-carbon atoms. Kundu *et al.* used the sign of the GNM first normal mode displacement, with some postprocessing, to assign residues to structural domains.[13]

A yet more challenging problem arises when only sequence features (but no structural coordinates) are known. In this article, we evaluate one predictor that uses only sequence information. Relatively little work has been reported on this largely unsolved problem,[14,15] but it is in some ways related to the more extensively studied problem of detection of evolutionary domain boundaries (which may or may not be flexible).[15–18]

In this work, we first introduce hNM, a family of mostly novel hinge predictors based on normal modes. The first member of this family, which we call hNMa for simplicity, posits that the minima of the normalized squared normal mode fluctuations should coincide with hinges. This in itself is not novel but we show that for the case of domain hinge bending, the first (rather than any higher) normal mode is most informative, addressing a point of some debate in the literature. A second, novel, method designated hNMb detects the most rigid, continuous structural domain through segmentation of normal mode motional correlation matrices. Subsidiary predictors hNMc and hNMd use similar information to find additional hinges. To benchmark the method and compare and integrate it with others, we use the Hinge Atlas Gold (HAG), a set of proteins with carefully annotated hinge locations.

We then turn our attention to existing methods, for purposes of comparison and integration. We review the following hinge predictors:

1. *StoneHinge* (Keating, *et al.* StoneHinge: A Hinge Prediction Algorithm Using Rigidity Theory. Manuscript in preparation, 2006) recognizes hinges as flexible regions of the protein main chain intervening between the two largest rigid domains (of at least 20 residues each), as defined by ProFlex constraint-counting analysis of the protein's covalent and noncovalent bond network. Importantly, StoneHinge has some ability to detect proteins that do not move by hinge bending, but rather fall into some other classification.[1] In the latter case, hinge prediction results from other predictors are likely to be inapplicable.
2. *Translation Libration Screw Motion Determination (TLSMD)*[19] divides the protein into segments whose rigid body motions best account for the observed distribution of temperature factors in a crystal structure.
3. The FO (*FlexOracle*[20]) family of hinge predictors generates protein fragments based on all possible locations of one or two cuts on the backbone. It is based on the idea that structural domains fold independently, therefore when the cuts coincide with the hinge location, the free energy of folding will be minimal for the corresponding fragment pair.
4. *HingeSeq*[15] is a hinge predictor based not on structure but rather on sequence features.

These methods are complementary to each other and to hNM family since they use very different information, namely bond network topology, experimental thermal factors, estimated domain free energy of folding, and sequence. We run all of the predictors against the HAG, then use various qualitative and quantitative measures to benchmark and compare the performance of each method. Lastly, we combine all of the methods using a voting scheme to create a new predictor called HingeMaster.

THE hNM FAMILY OF NORMAL MODE BASED HINGE PREDICTORS

hNMa: Which normal mode eigenvector is most important for hinge prediction?

We begin our development with the hNM family of hinge predictors. The name suggests its relationship to *HingeMaster* (the integrated hinge predictor) and its reliance on *Normal Mode* information. As mentioned, this family has five members, designated hNMa to hNMe, of which hNMb through hNMe are novel. The output of hNMa is simply the normalized squared fluctuations due to the first normal mode, with the idea that the minima of this quantity coincide with the hinge location. This in itself is not novel[21] but the choice of first versus second or higher modes is the subject of much debate in the community,[13,21,22] and it is this debate which we address first.

Normal mode expansions provide the form of displacements of a structure at each of a progressive series of resonant frequencies, or excitation frequencies to which an elastic structure responds strongly. Various studies underscore the importance of low-order modes in describing protein motion, but opinions vary as to which of these should be used for hinge prediction. Alexandrov *et al.*[23] and Krebs *et al.*[24] compared the successive normal modes of proteins with the displacements observed from interpolated ("morphed") structural pairs of proteins, and found that the correlation was highest for the lowest order mode, and decreased progressively for higher modes. Kundu *et al.*[13] assigned residues of protein to one of two clusters depending on the sign of the

lowest-order nontrivial normal mode eigenvector. These domain assignments are then adjusted by a series of physicochemically motivated postprocessing steps. Yang and Bahar showed that catalytic sites tend to coincide with regions of minimal displacement of the first and second nontrivial mode.[21] Here, we show that for the case of domain hinge bending, the lowest-order nontrivial mode should be used for hinge prediction.

To do so, we will make use of the concept of a nodal surface. To introduce this consider the example of a one dimensional guitar string driven at its second harmonic frequency. The string will have a nodal point in the middle which remains stationary. A drum head (effectively two dimensional) similarly will have nodal lines, depending on which mode is excited. A three dimensional object such as a tuning fork or a protein will have a surface which describes the locus of points that remain stationary when the object vibrates at one of its normal frequencies. The displacements of points on opposite sides of this nodal surface have opposite sign.[13] This surface is in some sense a hinge, about which the motion occurs.

One might come away from prior work,[12,23,25,26] with the following two *ideas*:

1. The nodal surface of the lowest order normal mode eigenvector should coincide with the hinge location.
2. The nodal surfaces of the second, third, and higher normal mode eigenvectors should also coincide with the hinge, but to a lesser degree.

To test these, we extracted the mobility score, M_{ik} for each residue i in the kth mode, for $k = 1$–7.[21] This quantity is the square fluctuation of residue i in mode k, normalized such that the most mobile residue has mobility $M_{ik} = 1$ for mode k. We then generated one ROC curve for each mode k. This is based on taking all residues with normal mode displacement lower than a certain threshold to be *test positives* and all others to be *test negatives*. The ROC curves are generated by moving the threshold and calculating sensitivity and specificity for each possible threshold. We found that the first idea above was correct; regions of low first normal mode displacement are likely to coincide with hinge location (see Fig. 1). The second and third normal mode displacements were not correlated with hinge location, as reflected by areas under the curve near 0.5. Modes higher than three were also found to have very little correlation with hinges (data not shown). Therefore the second idea is incorrect. From this we concluded that if normal mode displacements alone are used for hinge prediction then it is the first rather than higher modes that alone should be used. This is not to say that the higher modes are useless; in the next section we will show that the correlation matrix is generated by summing the correlations due to all modes, and this matrix can be used effectively for hinge prediction.

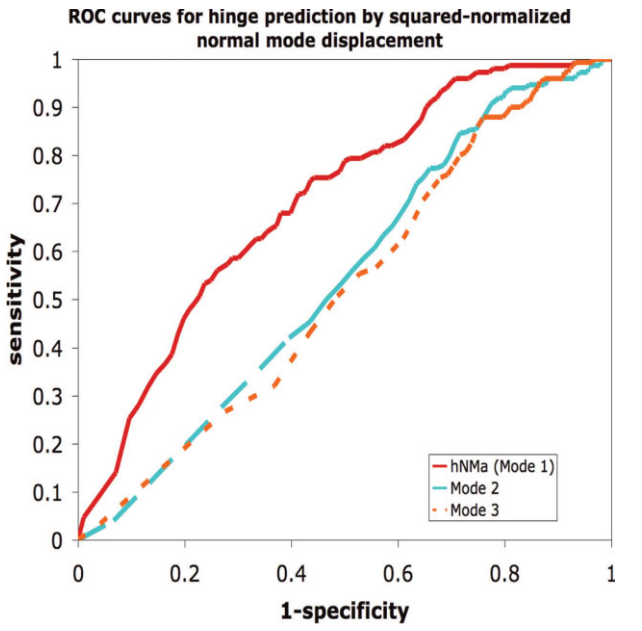

Figure 1

Lower values of the first normal mode displacement (solid line) correspond to hinges, as shown by an area under the curve significantly greater than 0.5. The same is not true of the second (dashed line) and third (dotted line) normal modes, which have almost no predictive power. The slope at the origin is not very steep, reflecting the fact that wide regions of proteins often had small or zero displacement, leading to ambiguous predictions. Regions of high displacement, however, were rather confidently excluded from containing hinges.

Motion correlation based methods (hNMb, hNMc, hNMd)

We now move on to describe applications of normal modes to hinge finding. To do this we must first calculate the normal mode motional correlations between α-carbon atoms in a protein. This is obtained by computing an expectation value in the Boltzmann ensemble. The result is[12,27]:

$$\frac{\gamma}{3k_B T}\langle \Delta R_i \cdot \Delta R_j \rangle = (\Gamma^{-1})_{ij} = \sum_{k=1}^{N-1} \omega_k^{-1} [u_k]_i [u_k]_j = U\Omega^{-1}U^T \quad (1)$$

Where Γ is the Kirchoff, or connectivity matrix, and Ω is the diagonal matrix of eigenvalues ω_k of Γ. The elements of Γ are simple to obtain approximately using the GNM method.[12] $[u_k]_i$ is the displacement of the α-carbon of residue i due to normal mode k. ΔR_i is the *net* displacement of the α-carbon of residue i from its equilibrium position due to normal mode motions. γ is the effective stiffness used by Tirion *et al.*[28,29] Cross-correlations are normalized with respect to the auto-correlations as follows[12]:

$$C(i,j) = \frac{\langle \Delta R_i \cdot \Delta R_j \rangle}{[\langle \Delta R_i \cdot \Delta R_i \rangle \langle \Delta R_i \cdot \Delta R_i \rangle]^{1/2}} \quad (2)$$

It is possible to inspect or analyze the cross-correlation matrix $C(i,j)$ by various methods[30] and determine the boundaries of domains. Figure 2(a) illustrates this—the continuous domain is easily identified visually in this case. The idea is that the motion of all the residues in a domain should on average be correlated with that of all other residues in the same domain. We therefore seek submatrices of $C(I,j)$ with high average values. Further, these submatrices should be maximal in size so as to favor finding the largest domains.

As a first step we compute the *Average Correlation* matrix as follows:

$$A(k,l) = \begin{Bmatrix} \frac{1}{(l-k)^2} \sum_{k<i\le l,\, k<j\le l} C(i,j) & \text{if} & k<l \\ 0 & \text{if} & k=l \\ A(l,k) & \text{if} & k>l \end{Bmatrix} \quad (3)$$

As mentioned we seek large domains. To this end, we therefore generate a matrix $W(k,l)$ by weighting $A(k,l)$ to favor pairs k,l that are maximally distant from each other, and are likely the endpoints of a structural domain:

$$W(k,l) = -|l-k| \cdot A(k,l). \quad (4)$$

Lastly, we treat $W(k,l)$ as a two-dimensional discrete function of k,l and identify its minima using the algorithm described for FlexOracle.[20] The $W(k,l)$ matrix, again for Glutamine Binding Protein, with its global minimum highlighted, is shown in Figure 2(b). hNMb (Continuous Domain Boundary Identifier), hNMc and hNMd (Excluded Region Identifier) all use this list of minima, but treat it differently. hNMb ranks the minima by the value of $W(k,l)$ at the minimum. The particular values of the indices k,l, where $k < l$ at the location of the global minimum are taken as the residue numbers of a pair of hinge points. If $k(l)$ is within five residues of the N(C) terminus, then $k(l)$ is dropped and the other index is reported as the sole hinge point. The last modification is that for the hinge point at $k(l)$ the hinge is reported as consisting of the two residues $k-1$ and $k(l-1$ and $l)$.

hNMc goes through the same procedure, except it ignores the lowest minimum (already reported by hNMb) and processes all remaining minima as above. If any hinge point is within five residues of a hinge point corresponding to a lower minimum, the hinge point corresponding to the higher minimum is discarded.

hNMd (Excluded Region Identifier) works somewhat differently. It is based on the idea that we may not be able to precisely identify the flexible regions of protein, but we can perhaps still identify parts of the protein which are rigid, and surmise that the hinge may lie anywhere *except* in these rigid regions. For a minimum of W located at residues k,l, it considers residues $k+1$ to $l-2$ to be part of a structural domain and excludes them from consideration as a hinge. The process is repeated with the remaining minima k,l of W (k,l). Any residues that were not excluded after all minima have been considered in this way are reported as potential hinges.

Lastly, hNMe (Holm and Sander-like hinge predictor) partitions the matrix differently from NMB with similar goals. The correlation matrix [Eq. (2)] is partitioned by separating the residues into two domains much as Holm and Sander[30] did for the contact matrix. A minor adjustment is needed which is explained in the Supplementary Methods section. This results in a reasonably good predictor (Table I) which has the added benefit of assigning each residue to one of two domains, and also has no intrinsic limitation with respect to number of hinge points. We did not use this method in the current work, however, since we found that the Continuous Domain Boundary Identifier (*hNMb*) described above yielded better results when measured by the associated *P*-values.

INTEGRATION OF HINGEMASTER

Gold standard

As we mentioned, in order to benchmark (and later combine) hNM and the other prediction algorithms we need a gold standard. In prior work, we found that hinge annotations reported in the literature are obtained by a wide variety of means, including NMR, proteolysis, secondary structure annotation, and normal modes.[20] Rather than attempt to treat such disparate annotations on an equal footing, in that work we compiled a set of proteins which had all been crystallized in two different conformations and exhibited hinge bending motion. This provides direct evidence of the structural conservation of domains through the course of motion. The hinges were then identified based on determining which residue backbone degrees of freedom needed to be free such that the observed conformational change could take place without large steric clashes.[15]

Many of the HAG proteins were collected from the Hinge Atlas, while others, such as Adenylate Kinase, Biotin Carboxylase, Lactoferrin, and Calmodulin, are classic hinge bending proteins widely used as test cases in the community and were added where absent to make the dataset represent proteins of broad interest as much as possible. The use of protein motions studied by third parties confirms that the annotated hinge points reflect biologically or at least thermodynamically relevant dynamics. We delve into this in greater depth for two proteins in this paper, and several more in the supplementary discussion. We will show that in some cases, we

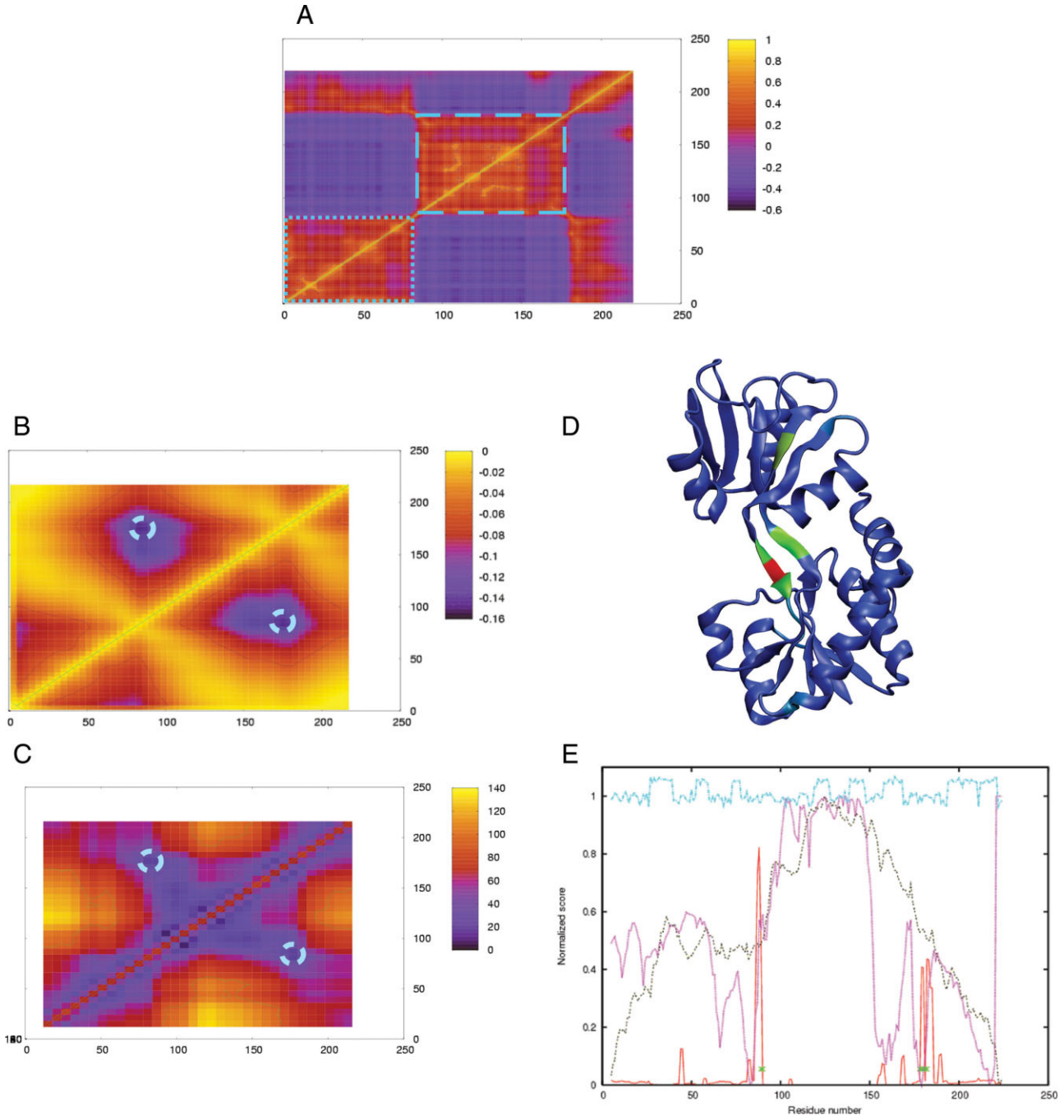

Figure 2

*Glutamine Binding Protein (open), Morph ID: f205132-23662 PDB ID: 1GGG, HAG hinges (residues 85–86,176–177). Above, (**A**) and (**B**) illustrate the procedure for generating hNMb and hNMd predictions. First, the α-carbon correlation matrix (plotted in a.) is obtained using GNM. In this case, residues ranging from 85 to 180 have highly correlated motion (dashed turquoise box). To a somewhat lesser degree, residues 1–80 are also correlated (dotted turquoise box). A second matrix is obtained the elements i,j of which contain the average correlation for a submatrix of the correlation matrix spanning residues i to j. This is then weighted by multiplying each element by $-|i-j|$. The resulting matrix is plotted in (b). The minima of this matrix correspond to the boundaries of structural domains which are continuous in sequence. The most significant of these (in absolute value) is i,j = 84,180, or equivalently 180,84 (dashed turquoise circles). A secondary minimum also exists at i,j = 1,80. hNMb therefore reports residues 84,85,180,181 as the predicted hinge location. hNMd works by excluding residues in continuous domains and reporting the remaining ones as possible hinges. Therefore, hNMd reports residues 85–88 and 181–217 as hinges. In (**C**) we show the two-cut FlexOracle (FoldX) plot. The minimum at 83,179 is visible (dashed turquoise circles). In (**D**) we show the protein colored by HingeMaster score. In (**E**) we show the output of the FO1, hNMa, HingeSeq, and HingeMaster predictors (dotted black, dotted magenta, dash-dotted cyan, and solid red traces, respectively) and the HAG hinge locations (green X's).*

Table I
Performance of the Predictors Against the Hinge Atlas Gold Annotation

c	Test positive	True positive	sensitivity	specificity	*P*-value
StoneHinge	1204	42	0.28	0.91	2.7E-11
HingeSeq	771	13	0.09	0.94	0.11
TLSMD	455	30	0.20	0.97	4.8E-15
hNMa	1279	37	0.24	0.91	8.7E-08
hNMb	126	31	0.20	0.99	3.2E-33
hNMc	517	26	0.17	0.96	1.7E-10
hNMd	1545	49	0.32	0.89	1.1E-11
hNMe	235	35	0.23	0.99	2.0e-29
FO1	563	8	0.05	0.96	0.32
FO1M	292	14	0.09	0.98	6.6E-06
FO	272	62	0.41	0.98	9.2E-66

Test positives are predicted hinge locations. hNM1 and FO1 give continuous (rather than discrete) output, normalized to range from 0 to 1 for each protein. Therefore for hNM1 and FO1 we took values below .02 and 0.1, respectively, as *test positives*. There were a total of 13,259 residues in the HAG, of which 152(13,107) were *Gold Standard Positives (Negatives)*. Therefore for the example of StoneHinge, sensitivity was calculated as 42/152 = 0.28 and specificity was (13259 − 1204)/13,107 = 0.91. For the same example, *P*-value was computed as the probability of finding 42 *or more* true positive residues in a set of 1204 residues selected *randomly and without replacement* from a set of 13,259, using the cumulative hypergeometric distribution.

predict a hinge where none is annotated in the HAG, but for which some evidence exists in the literature. In these cases that HAG annotation was not modified, since the point of the HAG is to be objective rather than comprehensive. These cases demonstrate that the predictors can detect previously unknown motion.

StoneHinge

We earlier introduced the *novel* hNM family of predictors, and now move on to describe the existing predictors. The first of these is StoneHinge, which predicts hinges using the FIRST algorithm[6] as implemented in the freely available ProFlex software. ProFlex is designed to analyze flexibility in proteins and uses a 3D constraint counting algorithm based on rigidity and graph theory. This approach ultimately derives from the structural engineering work of James Clerk Maxwell,[31] designed to assess whether the trusswork of a bridge would be adequate to ensure stability. The same concepts have been shown to be mathematically robust for analyzing the 3-dimensional covalent and noncovalent bond networks in proteins.[32] The FIRST algorithm[6,33] thus counts local degrees of bond-rotational freedom. This divides the protein into two types of regions: those that are constrained and therefore rigid, and those that are underconstrained and therefore free to rotate. A rigid region consists of a group of atoms that do not move relative to each other due to the constraints of the bond network. However, two rigid regions may move relative to each other, like two stones connected by a flexible tether.

A key part of this analysis involves representing the essential noncovalent interactions (hydrogen bonds, salt bridges, and hydrophobic interactions) that cross-bridge the protein structure.[8] These interactions vary in energy and can be separated by invoking an energy threshold. All interactions that are lower in energy (more favorable) than this threshold are included in the analysis, while those that are higher in energy (less favorable) than the threshold are ignored. To obtain flexibility analysis results reflecting "native-like" motion, this threshold is varied from protein to protein. We describe the process of selecting the correct threshold below.

StoneHinge builds on the FIRST algorithm and uses it to predict hinge motion. FIRST iteratively varies the threshold and examines the resulting bond-constraint network to find the boundaries of the rigid regions. The size, location, and number of the rigid regions will vary according to the threshold. StoneHinge selects the value of the threshold that results in a second-largest rigid region of maximal size. It then finds the flexible region or regions connecting the two largest rigid regions. These flexible regions are then reported as hinges.

StoneHinge typically requires that both rigid domains contain at least 20 residues. If two rigid domains of this size can not be found, then StoneHinge reports that any detected flexible residues are unlikely to contribute to a hinge bending motion, as domains of this size typically result from flexible loops or similarly small motions. However, this restriction was ignored for purposes of generating HingeMaster predictions.

Before the above analysis is carried out, StoneHinge performs some preprocessing steps as follows. It first removes all heteroatoms from the Protein Data Bank formatted structure files. These include inhibitors, ligands, and cofactors, as they may stabilize the protein in the ligand-bound conformation and cause the hinge region to no longer appear flexible. Additionally, it adds hydrogen atoms to the structure using GROMACS,[34,35] as they are required to calculate hydrogen bond energies, which are dependent on angular geometry as well as distance. Lastly, it removes all water molecules from input crystal structures. While ProFlex works best with only internal water molecules included, which can potentially be important for stabilizing protein structure, the effects of these waters are typically minimal.[36] As little difference was found in the hinge predictions when using vs. omitting entrained water molecules, (Keating, K., *et al.*, StoneHinge: A Hinge Prediction Algorithm Using Rigidity Theory. Manuscript in preparation, 2006) these were not included for this analysis. Further explanation and examination of the StoneHinge algorithm may be found in Keating *et al.* (StoneHinge: A Hinge Prediction Algorithm Using Rigidity Theory. Manuscript in preparation, 2006).

TLSMD

The second existing method was recently introduced (TLSMD version 0.8.1 was used here) to identify segments of a protein that exhibit concerted vibrational motion.[37] It is based on analysis of the spatial distribution of atomic displacement parameters within a single, conventionally-refined protein crystal structure. Each group so identified is modeled by assigning to it a set of 20 TLS (Translation/Libration/Screw) parameters that describe its net vibrational displacement.[38] The method is capable of identifying both large and small vibratory groups, and is largely independent of the resolution of the X-ray data used to refine the crystal structure. The validity of TLSMD analysis in the context of crystallographic refinement can easily be judged by tracking the standard crystallographic residuals R and R_{free} resulting from refinement of TLSMD-generated models against the original diffraction data. In many cases multigroup TLS models are strikingly successful at predicting the observed diffraction data, and out-perform conventional crystallographic models during refinement.[37,39] It is logical to conclude that TLSMD correctly identifies actual vibrational modes of the protein within the crystal. We are interested to test the extent to which these same vibrational modes are also present in solution, and the extent to which the boundaries between vibratory groups identified by TLSMD correspond to specific hinge points identified by other experimental methods.

TLSMD partitions the protein chain into N segments. It currently has no automated mechanism for distinguishing segment boundaries that might correspond to inter-domain hinges from segments boundaries that might, for example, define the endpoints of a flexible loop. To the extent that it assigns an order of importance to these boundaries, it does so based on their relative contribution to the observed distribution of atomic displacements in the crystal structure. Thus TLSMD may identify the boundaries of highly flexible, albeit small, regions before it identifies those of large domains whose vibrational amplitude is highly restricted by the crystal lattice. Therefore each segment boundary, or breakpoint, may possibly correspond to a hinge. But we faced a difficulty in deciding how many TLSMD breakpoints to compare against the hinge points listed in HAG. We have chosen to report all segment boundaries found for $N = 2,3,4,5$. This means that the hinges resulting from assuming 2,3,4, and 5 domains exist are all reported on an equal footing. We will delve into the consequences of this in the discussion.

FlexOracle (FO1, FO1M, FO)

The last of the existing algorithms to be discussed, FlexOracle, is based on the definition of structural domains as independently stable subunits of proteins. This means that cleaving the protein at the hinge site between domains should result in fragments that maintain their overall structure,[40] because those fragments should have a lower free energy of folding than fragments generated by cleaving within domains. The algorithm therefore works by estimating the free energy of folding for all possible pairs of fragments generated by cutting at two points on the protein chain, and looking for minima of this quantity.[20]

FlexOracle has powerful discriminating ability. Separate work describes three variants of FlexOracle.[20] These include the single-cut FlexOracle predictor with the FoldX force field, which we here call *FO1,* a second predictor which detects the local minima of the same, which we call *FO1M,* and the two-cut FlexOracle predictor, which here we simply call *FO.* FO is by far the most accurate of these as we will show.

Because it cuts the backbone at two points, however, FO is limited to proteins with single- or double-stranded hinges. Also, the code neglects bound metals. Since these are highly coordinated, we will argue that accuracy may be limited when metal binding contributes significantly to the stability and motion characteristics of the protein.

Definition of HingeMaster

As pointed out previously, the described algorithms use substantially different information to make hinge predictions. Consequently they have different strengths and yield very different results. StoneHinge is good at finding the general region of the hinge, but often overestimates the size of the hinge region. hNMd faces similar limitations. FO and hNMb are very precise but are limited to single or double-stranded hinges. TLSMD, on the other hand, makes a small number of predictions, well spaced apart, one or more of which often lie exactly on or very close to domain hinges, and the rest of which are incorrect or lie on points of non-domain hinge flexibility.

Combining various predictors by consensus[41] or other means is not unprecedented. We did this by creating HingeMaster, the output of which is a weighted vote of the individual predictors:

$$x_{\text{HingeMaster}}(i) = \sum_{\forall_c \in C} \lambda_c x_c(i) \qquad (5)$$

where

$$C = \{StoneHinge,\ FO1,\ FO1M,\ FO,\ HingeSeq,\ TLSMD,\ hNMa, hNMb, hNMc, hNMd, 1\}$$

$x_c(i)$ = output of predictor c for residue i.

λ_c = weighting coefficient of predictor c, determined below.

Parameterization of HingeMaster using Least Squares Fitting

Least Squares Fitting is a simple method used to project vectors onto a certain basis set, much as is done in a

Fourier Transform to project arbitrary functions into the basis of harmonic functions. We used the former technique to find the λ_c's in Eq. (5) corresponding to an optimal predictor. The procedure follows.

Let y = a column vector, the components $y(i)$ of which are the hinge annotations of the m residues in the HAG, in the format 1 = hinge, 0 = nonhinge. The index i counts over all residues in all proteins of the set in question, which in this work will be either the training, test, or complete HAG set. Order is unimportant as long as the *i's* in y are in the same order as the *i's* in x, below.

Let x = an m $\times$11 matrix, the rows of which will be used to predict the rows of y. Each column of x is a an m-component vector x_c, such that $c \in C$. Each component $x_c(i)$ of each such column vector is the output of the predictor c for residue i. Correspondingly, $x(i)$ (without a subscript) is a row vector with 11 components corresponding to the output each of the 11 predictors emitted for residue i. Note that the last "predictor" is a constant term used as an offset.

Let λ = a column vector, the components λ_c of which will give us the weight to be applied to the various predictors in order to make the composite HingeMaster predictor. Thus according to our definition of HingeMaster [Eq. (5)]:

$$x_{\text{Least-squares}} = x\lambda \approx y. \tag{6}$$

To obtain λ, we minimized the quantity $(x\lambda - y)^2$. The least squares regression methodology is a standard one[42] which will not be derived here. The result is that analytically:

$$\lambda = (x^T x)^{-1} x^T y \tag{7}$$

The above Eq. (7) can be said to *train* λ based on predictor output and gold standard annotation over some set of residues i. The best available value of λ is likely to be one fitted using the set of all residues in all proteins in the HAG, which we designate as {HAG}. That is to say, in Eq. (7) we use $x,y(i \mid i \in \{\text{HAG}\})$ and obtain a particular value of λ which we call λ^{HAG}.

Parameterization of HingeMaster using Boosting

Though simple, Least Squares Fitting results in a powerful predictor, as will be shown. We nonetheless also tried fitting the λ_c's defined above using an alternate machine learning technique called Boosting. This is a standard technique described in,[43] which we will briefly outline here. As for Least Squares Fitting, the goal of Boosting is to create a stronger classifier based on a series of predictors with individually weaker performance. In this setting, the outcomes of the hinge predictors are used as feature vectors of our learning algorithm, but instead of analytically minimizing $(x\lambda - y)^2$ as before, we iteratively minimize a loss function $\varepsilon(x\lambda,y)$ which decreases exponentially with classification accuracy.

Boosting is a generic technique with several variants, such as discrete AdaBoost which uses discrete predicted labels at each iteration of the algorithm. Real AdaBoost uses class probability estimates rather than discrete labels to improve accuracy. Other variants change the loss function to be Logistic and the gradient search method to be stochastic. For HingeMaster we use an extension of Real AdaBoost called Gentle AdaBoost.[44] It offers the advantage of using real valued class labels at each stage plus an improved numerical performance compared to Real AdaBoost.[45] The hinge residues comprise a small portion of total residues, resulting in an imbalanced dataset. Under these circumstances accuracy can trivially be maximized by a predictor which classifies all residues as nonhinges. To deal with this, we modify the method by oversampling the gold standard hinge residues 99-fold, with replacement.[46] Gentle AdaBoost is then run for 30 iterations after which $\varepsilon(x\lambda,y)$ converges to a minimum value. In the cross-validation that follows, we use the class probability estimates as a continuous predictor $x_{\text{prob\_Boosting}}$.

Cross-validating Least-Squares HingeMaster parameters

Clearly HingeMaster, whether trained using Least Squares Fitting or Boosting, cannot be tested on the same dataset used to train it. For both cases, we therefore validate HingeMaster by first randomly separating the 20 homologous pairs of proteins in HAG into a training set consisting of 15 of these pairs (30 total protein structures) and a test set consisting of the remaining 5 pairs (10 protein structures). The set of all residues in all proteins in the training set we call $\{TRAINING\}$, while the set of residues in the test set we call $\{TEST\}$. We used Eq. (7) with the *training set* data $x,y(i \mid i \in \{TRAINING\})$ to obtain λ^*, the cross-validation value of λ. We then used this vector with the individual predictor results for residues in the test set to obtain cross-validated HingeMaster results $x^*_{\text{HingeMaster}}(i \mid i \in \{TEST\})$ as follows:

$$x^*_{\text{HingeMaster}}(i \mid i \in \{TEST\}) = \lambda^* \cdot x(i \mid i \in \{TEST\}) \tag{8}$$

We then generated a ROC curve by gradually decreasing the threshold above which values of $x^*_{\text{HingeMaster}}(i \mid i \in \{TEST\})$ were taken to correspond to predicted hinge locations, and comparing these to the annotated hinge locations $y(i \mid i \in \{TEST\})$. For each value of the threshold, residues i with scores $x^*_{\text{HingeMaster}}(i)$ above that threshold are taken to be *test positives*. We further classify the test positives using a *strict criterion,* meaning that

Table II
Fitting of λ_c^{HAG} and λ_c^*

	x_c			λ_c^*		
c	Predicted hinge	Predicted non-hinge	λ_c^{HAG}	Average	Standard deviation	Description of c
1	1	1	0.042	0.062	0.010	Dummy constant for fitting
N	1	0	0.012	0.012	0.003	StoneHinge flexible regions
HingeSeq	high	low	0.004	0.004	0.001	Raw HingeSeq score; high values more likely hinge locations
TLSMD	0	1	−0.029	−0.048	0.009	TLS domain boundary, for N = 2.5 putative domains
hNM1	low	high	−0.010	−0.010	0.003	First normal mode displacement
hNMb	1	0	0.190	0.196	0.030	Continuous Domain Boundary Identifier most likely hinge location
hNMc	2,3,4..	0	0.028	0.035	0.007	Secondary hinge predictions similar to but not reported by the Continuous Domain Boundary Identifier
hNMd	1	0	−0.0007	−0.0006	0.0007	Regions excluded from contiguous domains
FO1	low	high	−0.008	−0.012	0.002	Single-cut FlexOracle (with FoldX) energy, normalized from 0 to 1. <0.05 = hinge
FO1M	1	0	0.014	0.015	0.008	Minima of single-cut FlexOracle energy, identified per Flores *et al.*
FO	1	0	0.189	0.165	0.022	Two-cut FlexOracle prediction

Values of c are given in the left column. The output x_c is given for predicted hinge and predicted nonhinge residues, for each predictor c. For example, hNMa gives output ranging continuously from 0 to 1, with the lower values more likely to correspond to hinge locations. hNMb, on the other hand, gives discrete output: 1 for predicted hinge locations and 0 for predicted non-hinge locations. Note that the sign of λ_c corresponds to whether higher or lower values correspond to hinges for that predictor. $c = 1$ is a dummy constant which compensates for the difference in mean values of predictors x vs. gold standard annotation y.

those that coincide exactly with annotated hinges ($y(i) = 1$) are taken as *true positives*, those that coincide with nonhinge residues ($y(i) = 0$) are taken as *false positives*, even if they are immediately adjacent to a hinge residue. The generation of the ROC curve is explained in more detail in prior work.[20]

We repeated the above process a total of 20 times. Each time, we generated new {*TEST*} and {*TRAINING*} sets by randomly dividing HAG as described. We obtained 20 different values of λ^* vectors and generated 20 different sets of HingeMaster predictions. These were then used to generate ROC curves representing average performance. We report the average and standard deviation of the HingeMaster parameters λ^* for Least Squares Fitting, where these values have an intuitive interpretation.

We also asked the question, what would happen if we modified our gold standard hinge definition to include five residues to the left and right of the HAG hinges? This is similar to the loose criterion, defined in the Statistical benchmarks section. Doing this would increase the number of True Positives at a given HingeMaster threshold, but would also annotate as "hinges" residues which clearly belong to a rigid domain. To answer this question, ROC curves were generated using the thus-widened HAG definition.

Cross-validating Boosting HingeMaster parameters

We underwent the process described above, with minor variations, to cross-validate the Boosting parameters. Instead of $x_{\text{least-squares}}$, we of course used x_{Boosting}. The {*TEST*} and {*TRAINING*} sets were generated in precisely the same way, but iteratively minimizing $\varepsilon(x\lambda, y)$ rather than $(x\lambda - y)^2$ resulted in different values of λ^* even for the same data. The sum in Eq. (5) was converted into a *class probability* $x_{\text{prob\_Boosting}}$ as mentioned. In every other respect the ROC curves were generated precisely as above, by varying the threshold value of $x_{\text{prob\_Boosting}}$ above which a residue was taken to be a hinge.

RESULTS

Weighting and evaluation of predictors

The fitting of λ^{HAG} and λ^* was carried out as described above. The averages and standard deviations of the resulting weighting factors are shown in Table II. We evaluate the predictors using the statistical measures of sensitivity (true positives/gold standard positives), specificity (true negatives/gold standard negatives), and *P*-value (see discussion) in Table I. Note that these were computed under the *strict criterion*, meaning that a test positive was considered to be a *false positive* if it coincided with a nonhinge residue, even if it was immediately adjacent to an annotated hinge residue. The statistical measures are explained in detail in the Supplementary methods section and in prior work.[20] The average ROC curves generated from the HingeMaster 20-fold cross-validation are shown in Figure 3. Four curves are shown, representing the two methods (Least Squares and Boosting), each trained and tested with two different gold standards (unmodified and widened HAG).

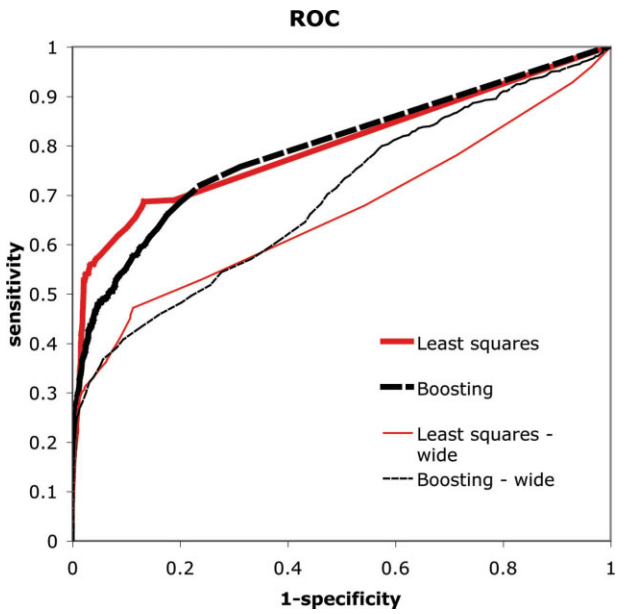

Figure 3

ROC curves representing average performance of HingeMaster in 20-fold cross-validation tests. Least Squares (thick continuous red line) had slightly greater Area Under the Curve (AUC) than Boosting (thick dashed black line), but the slope at the origin was slightly greater for Boosting. When instead of the HAG we used a gold standard which included five residues on each side of every HAG hinge (similar to our loose criterion), performance deteriorated (thin continuous red and thin dashed black lines represent Least Squares and Boosting, respectively).

Although the above summarizes the results of the various predictors, it is illustrative to review the results of the various predictors individually, for the 40 proteins in the HAG. We made an online gallery of results for that purpose at http://molmovdb.org/HAG. Links in this table for each protein in HAG lead to a morph page showing the motion between open and closed form of the protein, results of running the various predictors on the open and closed conformation, and the Protein Data Bank (PDB) information page for the associated PDB-deposited structure files. We also chose two of the forty proteins to discuss in detail here (Figs. 2 and 4 and below), and six more in the supplementary information.

Glutamine binding protein (GlnBP) (open)

Morph ID: f927198-20246 PDB ID: 1GGG
HAG hinges (residues 89,90,178–182)

Examination of results for individual proteins can bring out salient features of the various predictors. As a first example, we present Glutamine binding protein (GlnBP). GlnBP of *E. coli* resides in the periplasmic space, where it binds L-glutamine and subsequently undergoes a conformational change that allows it to be recognized by the membrane-bound components of the permease system, which subsequently translocate the nutrient into the cytoplasm against a concentration gradient.

GlnBP is comprised of two domains linked by a hinge region, the approximate location of which was arrived upon by various authors using different methods. Pang *et al.* annotated the hinge location by taking two extreme projections of the protein coordinates along the second normal mode eigenvector, and then used Hingefind to identify a hinge point at residues 88 and 183, based on these two structures.[47] This is very different from the HAG annotation procedure, which used two different crystallographically obtained conformations of the protein. The so-named large domain contains both the N- and C-termini and consists of two stretches of polypeptide, residues 1–84 and 186–226 according to Hsiao *et al.*[48] The small domain therefore consists of a single stretch of polypeptide from residues 90–180. Hsiao *et al.* simply took the entire PROCHECK[49]-identified antiparallel β-stranded region connecting the two domains (residues 85–89 and 181–185), as a hinge, again referring to only one structure.

hNMb, and hNMd [Fig. 2(a,b) and 5(c)], and FO [Fig. 2(c) and 5(c)], were successful, but hNMd overpredicted the extent of the hinge region. Stonehinge fragmented the largest domain into several smaller domains [Fig. 5(c)]. Thus, its predictions correspond to a flexible loop in the second domain. The StoneHinge output noted that this prediction likely did not correspond to domain motion; however, as explained above, this note was ignored for the HingeMaster predictions.

The TLSMD partition into 3 chain segments for 1GGG:A (Chain A) is quite accurate, placing the segment boundaries at 88/89 and 182/183 [Fig. 5(c)]. For the second chain in the structure, 1GGG:B, TLSMD instead finds a boundary at 168/169 in the 3-segment split. The TLSMD analyses of both chains A and B agree on $N = 4$ boundaries at residues 90 ± 2, 169 ± 1, and 190 ± 1. The false positive boundary at 44/45 seen in this figure is introduced when a 5th chain segment is requested.

hNMa shows global minima near the HAG hinges, while FO1 is less clear about this. HingeMaster (with Least Squares fitting) displays very clear peaks at or very near the HAG hinges [Fig. 2(e)]. The "color by HingeMaster flexibility" feature available on our server is illustrated in Figure 2(d), where strong hinge predictions can be seen to lie on the linker connecting the two domains.

Human lactoferrin

Morph ID: f964647-15593 PDB ID: 1LFG
HAG hinges: 90,91,250,251

Human lactoferrin (hLF) is an iron-binding glycoprotein found in exocrine fluids produced by mammals, including milk, saliva, tears, bile, pancreatic fluid, and mucous secretions. It has broad spectrum antibacterial

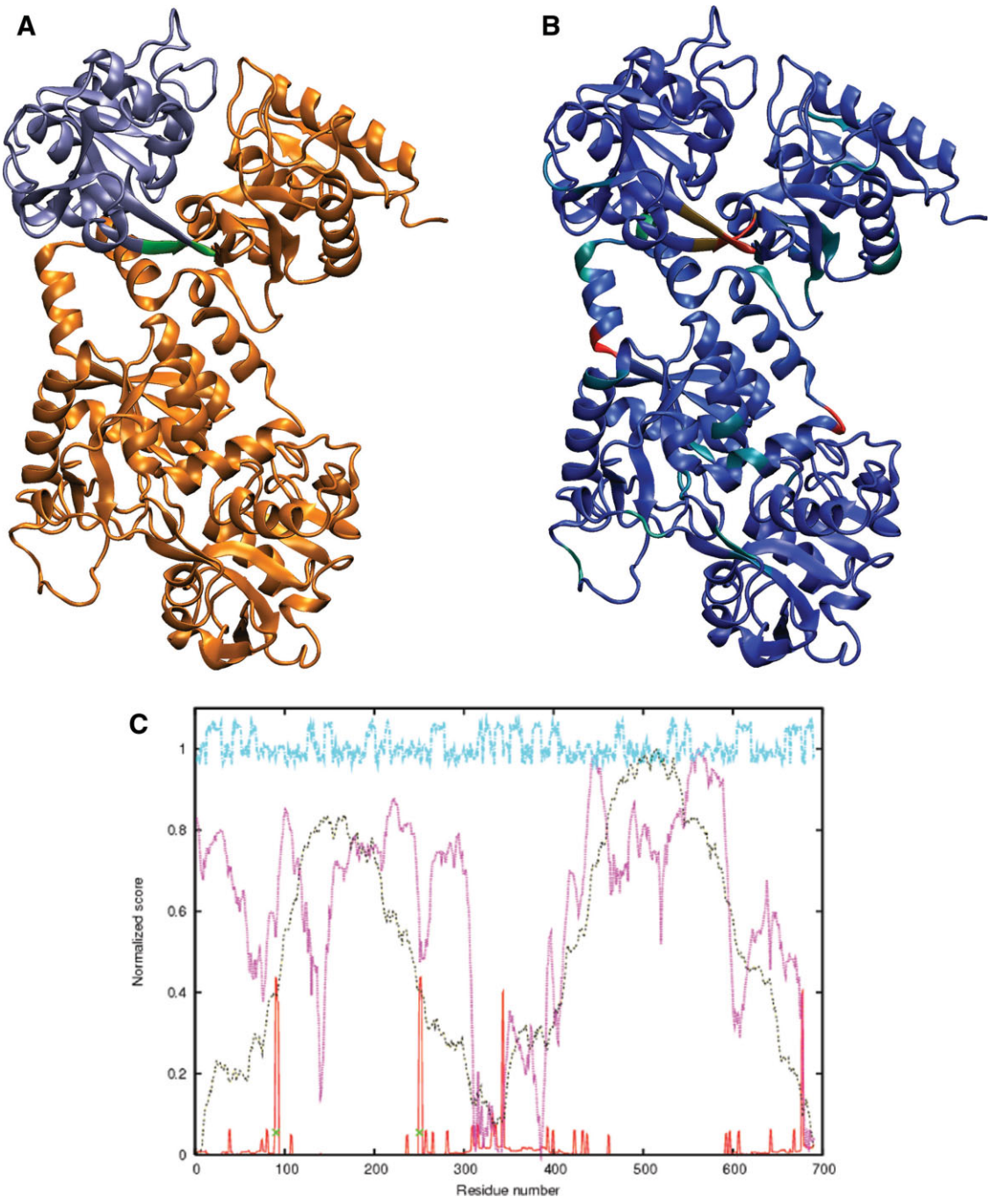

Figure 4

*Human lactoferrin (open, apo form), Morph ID: f964647-15593 PDB ID: 1LFG, HAG hinges: 90,91,250,251. In (**a**), the HAG hinge is shown in green, with domains on either side colored differently. As seen in (**b**), HingeMaster makes a strong prediction for a hinge at residue 332 (four algorithms in agreement, **C**, see Figure 2 caption for legend). This location does not appear in the HAG, but a search of the literature uncovered evidence for a hinge here. Lactoferrin is organized into two homologous halves, named the N and C lobes. Each of these lobes is further subdivided into two domains: N1 and N2 in the N lobe and C1 and C2 in the C lobe. The opening of these domains exposes an iron binding site in each lobe.[52] The motion selected for Hinge Atlas Gold shows the N1 domain (shown in blue) rotating relative to the rest of the molecule exposing the binding site in the N-lobe. Thus, the HAG hinges are located between the N1 and N2 domains. The hinge most strongly predicted by HingeMaster, however, falls in between the C1 and N2 lobes and contributes to the movement of the two lobes relative to each other. As in the case of Troponin C, experimental evidence is found for this hinge, as lactoferrin crystals grown at 277 and 303 K show rotation between the two lobes (Karthikeyan et al.). None of the predictive measures identify a hinge point at 90/91 in the open form structure. However TLSMD analysis of the closed form of the same protein (PDB entry 1LFH; Morph ID f964705-18231) finds segment boundaries at 91/92 and 249/250 for the 3-segment partition, corresponding exactly to the lactoferrin mobile domain boundaries (not shown).*

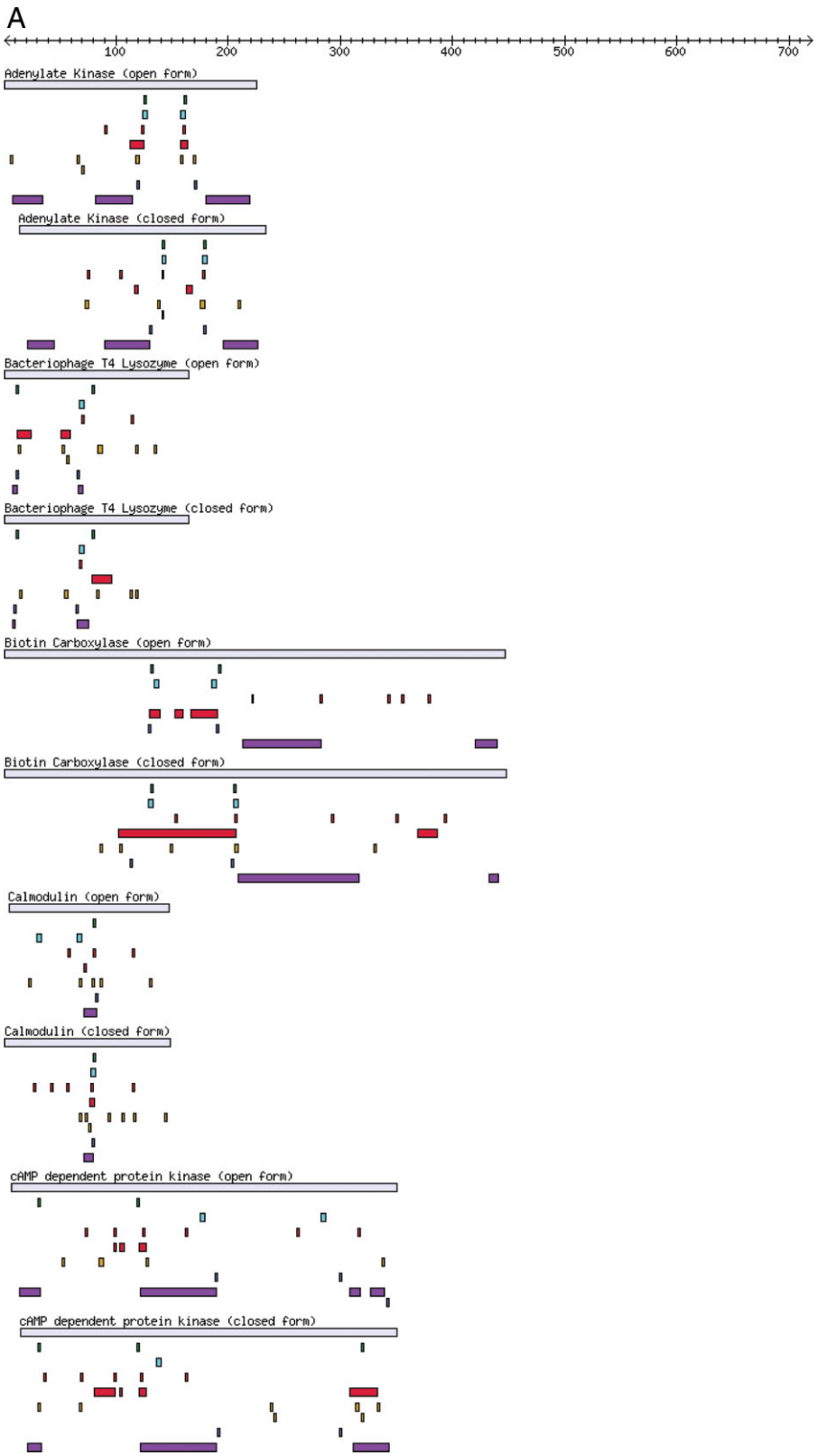

Figure 5

Predictor results for HAG proteins at a glance (panels a-d). Rule at top indicates residue number along protein chain. Each protein is marked with protein name. Light blue track represents residue numbers spanned by each protein. Green boxes indicate location of HAG hinge. Cyan, magenta, red, orange, blue, and purple bars indicate hinge location predicted by FO, FO1M, StoneHinge, TLSMD, hNMb, and hNMd, respectively.

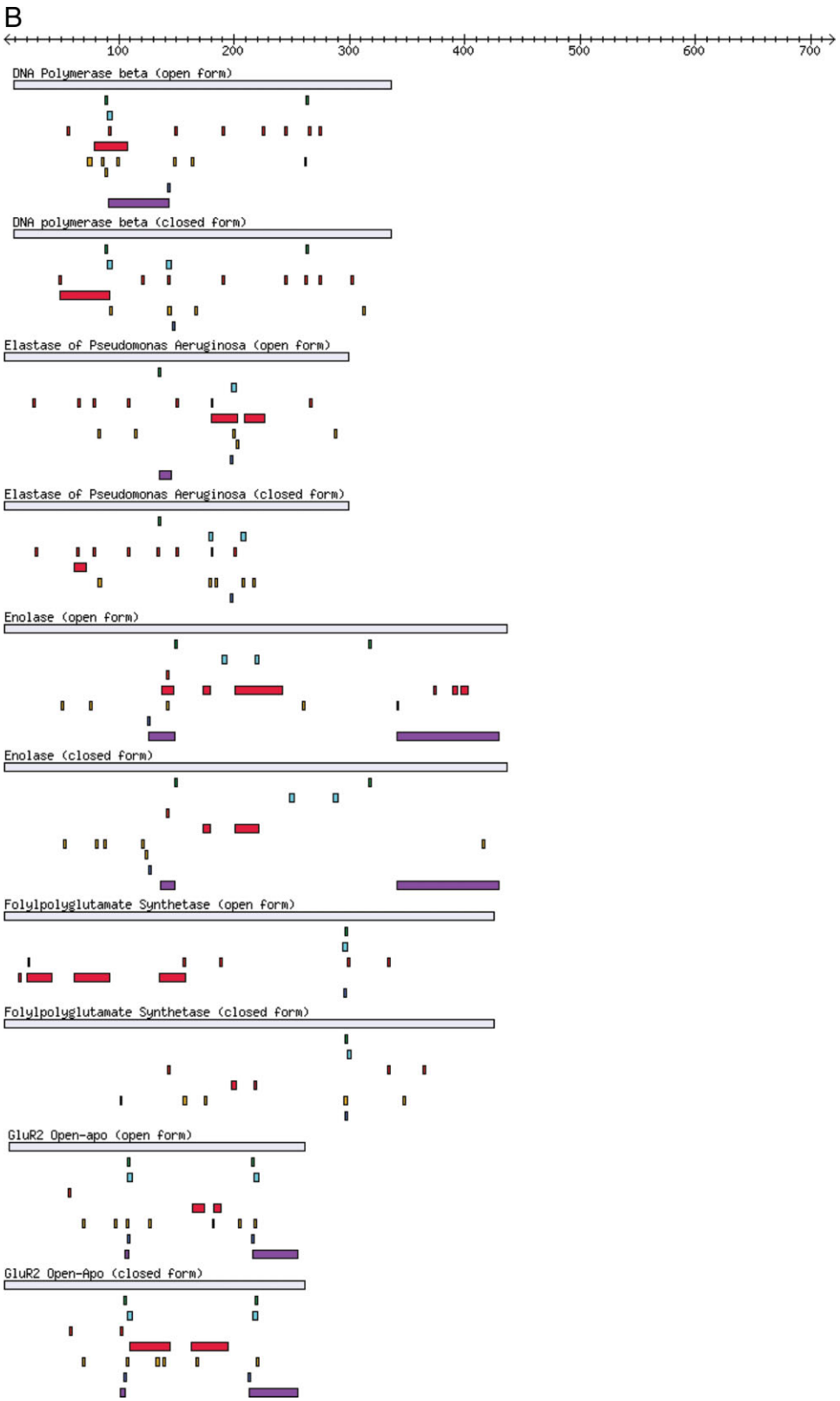

Figure 5
(Continued)

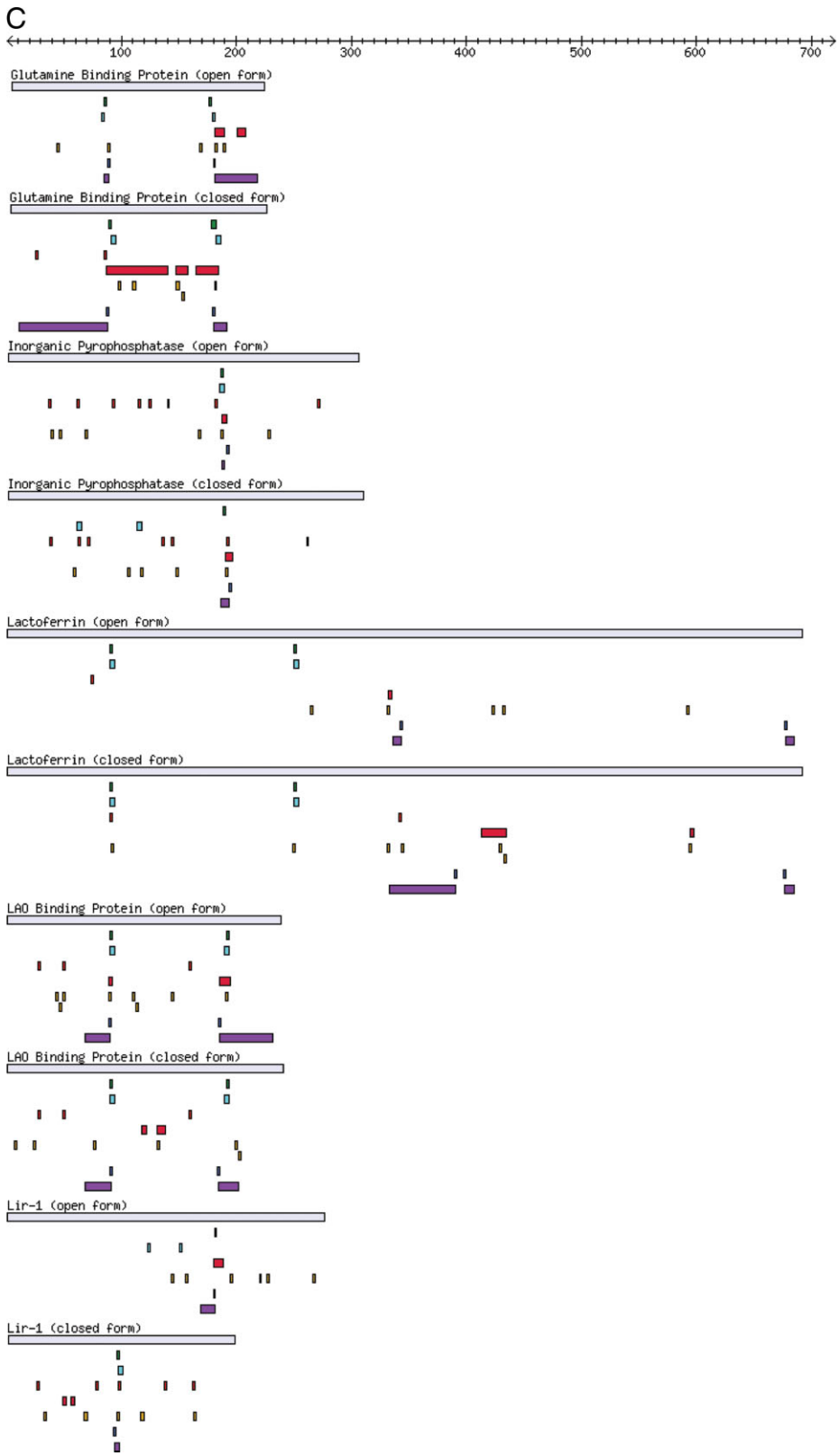

Figure 5
(Continued)

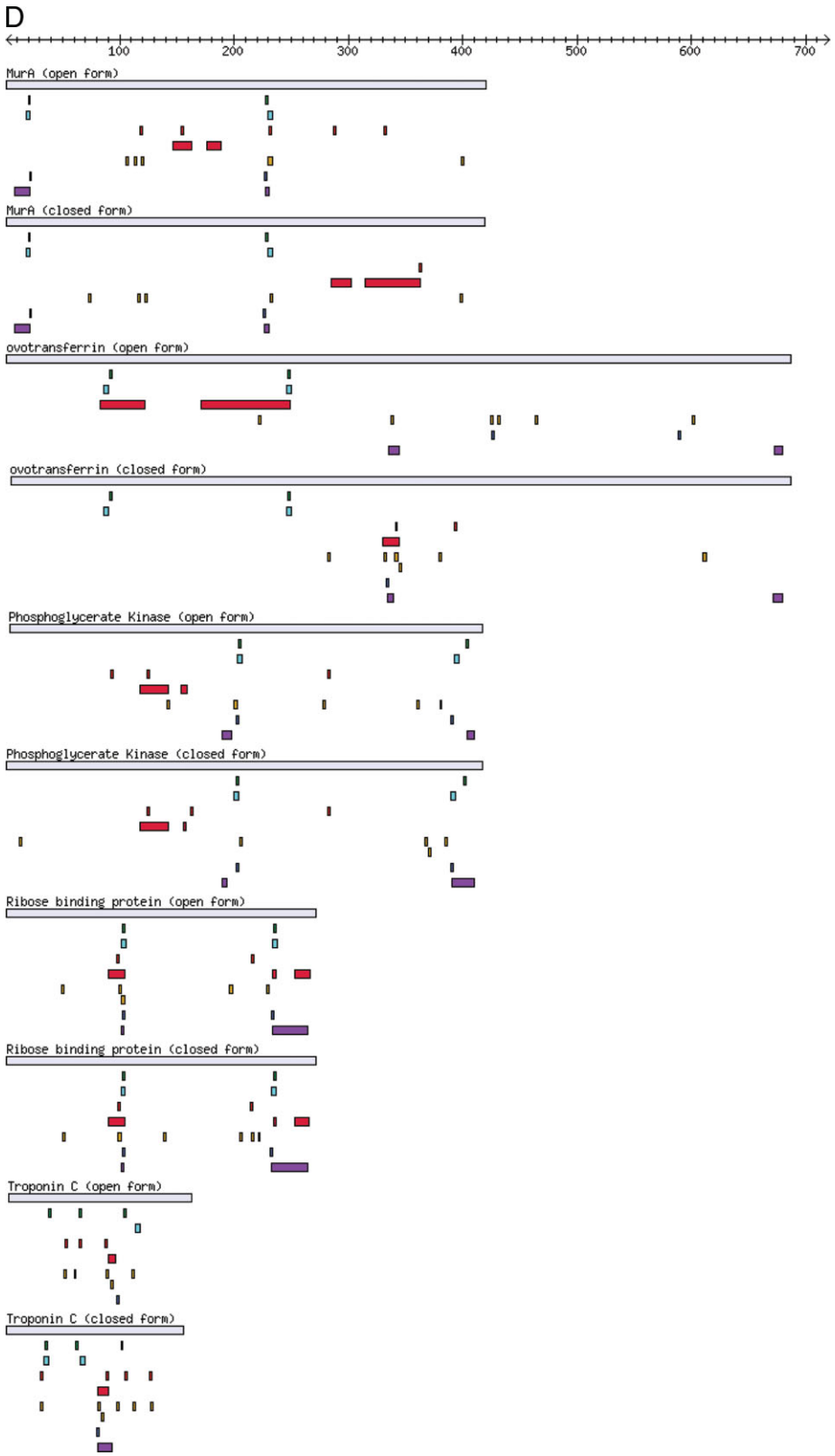

Figure 5
(Continued)

properties and seems to regulate the absorption and excretion of iron in infants.[50]

The hinge bending motion of lactoferrin has been studied in detail. The protein consists of N- and C-terminal lobes which are highly homologous and are presumed to have arisen from gene duplication. Each lobe is further subdivided into two domains, N1 and N2, and C1 and C2. In the iron-free form, a deep cleft appears between N1 and N2. No such cleft appears in the C-lobe either in the iron free or iron bound form, but this is believed to be an artifact of crystallization. In the iron-bound form, N1 and N2 are close together about a common hinge, located between residues 90 and 91, and 250, and 251 according to Gerstein *et al.*[51] This is in perfect agreement with the independent annotation made in this work.

The results for this protein are shown in Figure 4. HingeMaster makes a strong prediction for a hinge at residue 332 (StoneHinge TLSMD, hNMb, and hNMd in agreement). This chain location does not appear in the HAG, but a search of the literature uncovered evidence for a hinge at this location. Lactoferrin is organized into two homologous halves, named the N and C lobes. Each of these lobes is further subdivided into two domains: N1 and N2 in the N lobe and C1 and C2 in the C lobe. The opening of these domains exposes an iron binding site in each lobe.[52] The motion selected for HAG shows the N1 domain (in blue) rotating relative to the rest of the molecule exposing the binding site in the N-lobe. Thus, the HAG hinges are located between the N1 and N2 domains. The hinge most strongly predicted by Hinge Master, however, falls in between the C1 and N2 lobes and contributes to the movement of the two lobes relative to each other. As in the case of Troponin C, experimental evidence is found for this hinge, as lactoferrin crystals grown at 277 and 303 K show rotation between the two lobes (Karthikeyan *et al.*).

None of the predictive measures identify a hinge point at 90/91 in the open form structure. However TLSMD analysis of the closed form of the same protein (PDB entry 1LFH; Morph ID f964705-18231) finds segment boundaries at 91/92 and 249/250 for the 3-segment partition, corresponding exactly to the lactoferrin mobile domain boundaries (not shown).

DISCUSSION

After completing the calculations, we carried out our analysis of the hinge finding algorithms on two levels: the quantitative level, which consisted of examining various statistical benchmarks, and the qualitative level, which consisted of interpreting results for individual proteins. In this section we will discuss the quantitative summaries and will also point out advantages and disadvantages of the algorithms through the specific examples presented in here and in the supplementary information. In this section as before we will focus on StoneHinge, TLSMD, hNMb, and FO, and to a lesser extent hNMd, only intermittently referring to the various other subsidiary predictors and HingeSeq.

Statistical benchmarks

We begin to get an idea of the predictive value of the various methods from the weight assigned to each predictor, in Table II. Here, we see that hNMb and FO receive by far the highest weights, hinting at high predictive power, while TLSMD and StoneHinge get a more moderate weight and hNMd gets almost no weight. This cannot be taken as a rigorous statistical benchmark, however, since the predictors are not independent, and some, such as FO and FO1M, are in fact very closely related. For greater rigor, we computed the sensitivity, specificity, and *P*-value associated with each predictor in Table I. These statistical measures are explained in detail in prior work.[20]

The sensitivity (true positives/gold standard positives) was highest for FO, followed by Stonehinge, TLSMD, and hNMb. FO was thus able to find the largest number of hinge residues (62). As is customary we also report the specificity (true negatives/gold standard negatives). hNMb had the highest specificity, followed closely by FO and TLSMD. Stonehinge and hNMd have lower specificity, reflecting a tendency to predict the correct hinge location but also to report a wide region about the annotated hinge location as a predicted hinge. We also evaluated the statistical significance by postulating a *null hypothesis* that the test positive residues were taken randomly and without replacement from the population of residues in HAG. Under this hypothesis, the mean frequency of hinges is the same in the test positive set as in HAG. For each predictor, we computed the probability of finding the observed number or more of annotated hinge residues in a randomly selected set equal in size to the number of test positives reported by the predictor. We computed this quantity using the cumulative hypergeometric distribution.[20] Clearly, FO has the highest statistical significance (lowest probability that the null hypothesis is correct). hNMb has a higher (but still impressive) *P*-value than FO.

Part of the reason FO had lower specificity than hNMb is because FO only attempts cuts on the backbone at four residue intervals, while hNMb probes every possible pair of residues as a possible continuous domain boundary. As a result, FO predicts four-residue wide windows as possible hinge locations, with some uncertainty as to the exact location of the best putative hinge. hNMb, on the other hand, reports two-residue windows as putative hinge locations, and these are presumed to be the best choice of hinge possible for that method.

Table III
Summarized Performance Under the Loose Criterion

	StoneHinge	TLSMD	hNMb	hNMd	FO
Successes	8	0	17	19	23
Failures	20	12	14	13	12
Partial successes	12	28	9	8	5

With regard to the *P*-value, the reader should recall that this quantity changes based on the size of the dataset. Therefore this number can only be used to compare predictors tested on the same data. By this measure, FO was by far the most discriminating predictor, followed by hNMb.

As explained earlier, the sensitivity, specificity, and *P*-value were computed under the *strict criterion* for statistical rigor. However, for an application most users would probably consider a prediction that came within about five residues of the correct hinge location to be a true positive. So we examined each of the 40 proteins and labeled each as a success, partial success, or failure on the basis of a loose criterion for each predictor *c*. A test was considered a *success* under this criterion if each prediction came within five residues of an annotated HAG hinge, and vice versa. It was a partial success if at least one prediction matched one HAG hinge, but one or more predictions were more than five residues from a HAG hinge, or vice versa. It was a failure if no predictions were within five residues of a HAG hinge. The evaluation for each protein under the loose criterion is presented in supplementary Table V.

We counted the number of successes, partial successes, and failures for the five most interesting predictors in Table III. As can be seen, FO scored the most successes, followed by hNMd and hNMb. TLSMD had no successes, but this is due to the fact that as implemented on our server it reports all domain boundaries for up to five domains. Since the proteins in HAG have at most three hinges, some of the TLSMD predictions would be expected to have no corresponding HAG hinge.

FlexOracle (FO)

The FlexOracle algorithm has the best predictive ability by several measures. It had the highest sensitivity and lowest *P*-value (Table I). It had the most successes under the loose criterion (Table III). Results for individual proteins (supplementary Table V) show that FO failed for Calmodulin, Troponin C, and Elastase of Pseudomonas Aeruginosa, suggesting the method cannot deal well with proteins that depend on bound metals for stability and motion. It also fared poorly for the two proteins with three hinge points, as would be expected for a predictor intrinsically limited to two hinge points (supplementary Table V).

hNMb, hNMd

hNMb and hNMd are complemented by FO, as can be seen by several cases for which FO fails but hNMb and/or hNMd succeed (Fig. 5), as was the case for five proteins: the (open) metal-bound form of calmodulin, bacteriophage T4 lysozyme (closed), cAMP dependent protein kinase (closed), elastase of pseudomonas aeruginosa (open), and inorganic pyrophosphatase (closed).

The hNMd method has a strong advantage in that unlike FO and hNMb, it is not intrinsically limited by the number of strands in the hinge. In particular, we note that FO and hNMb both completely failed to predict the triple stranded hinge in cAMP dependent protein kinase, while Stonehinge and TLSMD either failed or had partial success. For the closed form of this protein hNMd was the only successful predictor. For the open form, one of the three hinge points is in a disordered region that does not appear in the crystal structure. hNMd has a cluster of hinge predictions centered about that break in the chain and therefore it was arguably as successful as could be expected under the circumstances. The same argument could not be made for the other predictors, as the reader can verify by examining Figure 5.

In general, we find that residues identified by hNMd coincide to a high degree with the general location of the hinge. However the predicted hinge regions were broad, lowering the specificity and raising the *P*-value.

TLSMD

As mentioned earlier, because hinges for $N = 1,2,3,4,5$ are reported together, the set of TLSMD "hinge predictions" for each protein will be larger than the actual number of hinge points. That is, the forced partition of a 3-domain protein into only two segments will tend to create a "false" breakpoint somewhere inside the middle domain, rather than finding either of the two "true" domain boundaries. Even if the two correct boundaries are found by the TLSMD partitions for $N \geq 3$, this initial false prediction will remain in the prediction set. This results in a poorer showing on a ROC curve if the extra predictions are treated as false positives. An alternative would have been to manually filter out these extra breakpoints based on visual inspection of the structure, but we were reluctant to introduce possible experimentor bias. On the basis of the outcome of the evaluations presented here, and on the observed distribution of actual hinge points in the larger set of TLSMD breakpoints, we hope to be able to automate such filtering in the future. To a significant extent, however, this filtering is done by HingeMaster. The StoneHinge, hNMa, hNMd, and FO1 predictors, which have many predictions and low specificity,

Table IV
Predictors at a Glance

Predictor	Basis	Max. hinge points
FlexOracle	Free Energy of folding	2
hNMb	Normal modes	2
hNMd	Normal modes	No limit
TLSMD	Experimental thermal factors	No limit
StoneHinge	Bond network topology	No limit

Three of the predictors have no limit on number of hinge points *in principle* but on our server TLSMD only reports boundaries based fragmenting the protein into as many as to 5 domains. Domain motions with a very large number of hinges are unlikely for proteins of the size range considered here.

tend to "select" the TLSMD breakpoints corresponding to domain hinge motion, while "deselecting" those corresponding to fragment hinges or other motions.

We found that some of the hinges predicted by TLSMD were close to HAG hinges with high frequency. The remaining TLSMD predictions often correctly reflected motions of fragments smaller than domains, and these tended not to coincide with the predictions of other methods. An additional stage of either manual or automated curation of the TLSMD results would remove most of these fairly easily. In particular, TLSMD would benefit greatly from an improved ability to combine multiple chain segments from the initial analysis to yield a description of 'domains' in the usual sense. The MurA analysis in the supplements provides a good example of this. The essential features of the structure are captured well by the TLSMD partition into six continuous chain segments. The close three-dimensional proximity of the boundaries at residues 20/21 and 228/229 are easily interpreted as belonging to a single inter-domain hinge; the further subdivision of the obvious continuous domain into three chain segments is easily interpreted as the presence of a small flexible loop (residues 108–127) protruding from a larger continuous domain. The C-terminal tail of ~20 residues is also flexible, but is not relevant to the primary inter-domain flexibility. Thus a curated interpretation of the TLSMD analysis of MurA would list three features: A two domain protein with an inter-domain hinge points at residues 20/21 and 228/229; a secondary set of hinge points corresponding to a flexible-loop (108–127) in one of the domains; and a flexible C-terminus.

StoneHinge

We discussed the fact that TLSMD overpredicts, since it reports more hinges than are annotated in HAG. StoneHinge also had a significant rate of false-positive hinge predictions (Table IV), This is typically caused by StoneHinge underpredicting the size of the rigid domains, such as in the open form of ovotransferrin. In that case, StoneHinge predicted that the second domain spanned residues 121–171. However, observation of the motion reveals that this domain spans residues 93–245. This underprediction of the rigid domain leads StoneHinge to overpredict the span of the hinge. Additionally, a number of StoneHinge's over- and mis-predictions are flagged as such by StoneHinge. As noted above, if either rigid domain is predicted to be less than twenty residues, StoneHinge reports that the predicted hinges are unlikely to correspond to domain motion (again, this eventuality is ignored for HingeMaster).

We note that when both TLSMD and StoneHinge predict a hinge, two-thirds of the time it coincides with a HAG hinge. Likewise, most experimentally defined hinges are predicted by either StoneHinge or TLSMD, accounting for the substantial weight they receive in HingeMaster. In the cases where either StoneHinge or TLSMD misses a hinge, it is usually predicted by at least one of the other predictors. Notably, experimentally defined hinges virtually always occur within the lower 10% of the hNMa atomic displacement function. Accordingly, the predictive confidence of HingeMaster is strong when at least three of these methods predict a hinge at a given site (or within a few residues of it). As with computational hinge detection, experimentally defined hinges can be identified based on different criteria. For instance, one researcher might identify them based on the observation of large, localized, noncompensatory changes in main-chain dihedral angles, whereas another might identify hinges based on large main-chain B-values (which can reflect rigid-body motions of a well-ordered structure in one case, but a lack of well-defined local structure in another). Thus, there is a distinct possibility that the various hinge predictors are identifying hinges that represent different mechanisms of motion, especially with regard to how localized or disseminated that motion is. By defining different hinge mechanisms and algorithms for detecting them, we hope to ultimately clarify the kinds of motion that occur in proteins, and provide tools that will aid in annotating experimental structures.

Least squares versus Boosting, and width of hinge region

The Least Squares method of training HingeMaster resulted in a powerful predictor, as demonstrated by a ROC curve with large Area Under the Curve (AUC) and high, nearly vertical slope at the origin. We nonetheless investigated whether a more sophisticated algorithm could improve results. Despite its additionally complexity Boosting was found to yield results only marginally different from those of Least Squares fitting (see Fig. 3).

One must bear in mind that the hinge residues are a small portion of total residues, therefore the Gold Standard Positives used for training are a very small set. Under these circumstances, classifiers with many adjustable parameters suffer from the problem of overfitting,[53] and

simpler methods are appropriate. The version of HingeMaster offered on our server and discussed in most of this work was trained using the Least Squares method.

We also sought to determine whether HingeMaster would benefit from training with a gold standard that included not only the HAG hinges, but in addition all residues within five residues of the nearest HAG hinge, similar to the loose criterion. This is a much more forgiving definition of the hinge. However the gold standard thus generated included residues which clearly belonged to one or another rigid domain and so was of lower quality. Correspondingly, the results were found to deteriorate for Least Squares as well as Boosting, as evidenced by ROC curves with significantly lower AUC (Fig. 3).

Complementarity of methods

Since the various predictors are based on very different information (Table IV), their strengths and weaknesses are complementary. In particular, FO and hNMb are limited to single- and double-stranded hinges. hNMd, however, has no such limitation and may be more successful in these cases (supplementary Table V). Similarly, FO is susceptible to bound metals that play a significant role in stabilizing the protein, but hNMb and hNMd is reasonably successful in these cases. TLSMD is comparatively weak at distinguishing domain from non-domain motion, but can find smaller scale motions not detected by FO and hNMb, and its accuracy is unaffected by bound metals. Stonehinge lacks precision but is good at finding the general region of the hinge. Altogether, only a few hinges escaped detection by one or another of the methods. The combination of highly specific predictors which usually did well but were sometimes somewhat off the mark, with predictors which identified broad swathes where the hinges were likely to be, resulted in a predictor of variable sensitivity, as we will discuss. The output of HingeMaster strongly indicates the pinpointed location of the hinge predicted by FO and hNMb, but less dramatically points out alternative locations. When interpreted by a critical eye, these results could bring insight even when one or more of the predictors are incorrect. Some of this is visible in the results for individual proteins as discussed.

CONCLUSIONS

We demonstrated the strengths and weaknesses of several predictors, including a set of normal mode based tools. We show that for 29 of the 40 proteins in HAG at least one predictor is completely successful under the loose criterion. For 10 of the remaining 11, at least one predictor was partly successful. HingeMaster weighs the correlation of each predictor to the HAG hinge annotations and presents the combined results in a visually understandable way. This combined predictor is shown to robustly produce ROC curves demonstrating high predictive power.

WEB TOOLS

Hinge prediction server

Most of the predictors discussed here can be run by the public on single-chain proteins by making a submission through the hinge prediction server linked from the front page of our server, molmovdb.org. When the job is complete, the user receives an email with a link to the generated morph page. The "Hinge Analysis Tools" box has links to output from the five predictors. The most useful of these is the "Combined predictor page," which shows the results of all analyses in a single graph, as was done for this article. Also, buttons are available to highlight the hinges predicted by HingeMaster, TLSMD, and StoneHinge in the jmol window. It is possible to color the protein by HingeMaster flexibility, as we will describe below.

Hinge annotation tool

To manually annotate the hinges in a submitted protein, one can use the Hinge Annotation Tool in the hinge analysis toolbox, on the morph page. This tool consists of three rows of arrow buttons which allow for the selection of up to three hinge locations. A "?" button on each row returns the residue number of the current selected hinge location. A "Show all" button highlights all selected hinge locations. A "Reset highlighting" button returns to the default view. The "Submit" button must be clicked for these annotations to be entered into our database. There will appear a "display public hinge" button which will allow all users to view the selected residues in the jmol window. With minor modification, this tool was used to annotate the HAG hinge locations used in this work. It is possible to create rendered images with hinges highlighted as we explain below.

Render studio

As discussed above, high resolution "domain style" images similar to those in the figures can be generated by the public by following these steps:

1. Select up to three hinge points using the Hinge Annotation Tool, as described above. Click on the "Submit" button.
2. Orient the molecule to the desired perspective by clicking and dragging in the Jmol window.
3. Click on the "color by domain" link.

The coloration of the domains goes by the following logic. All the residues prior to the first hinge point are

assigned to domain D1, all the residues between the first and second hinge points belong to D3, all the residues between the second and third hinge points belong to D1, and all subsequent residues belong to D3, and so on. The hinge residues themselves belong to D2. D1 is colored orange, D2 is green, and D3 is blue.

To color by HingeMaster flexibility step 1 above is unnecessary; in step 3 click instead "color by HingeMaster score." In either case after a slight delay, a pop-up window will display the generated image. A variant of this tool was used to generate some of the images in this paper.

ACKNOWLEDGMENTS

Authors thank Cheryl Leung for extensive help with the preparation of this manuscript, Nat Echols for helping with the web server, Mihali Felipe for systems administration help, and Burak Erman and Xin Chen for illuminating mathematical discussions. Parts of this work were discussed and refined at the workshop "Dynamics under Constraints" at Bellairs Research Institute of McGill University, Holetown, Barbados, January 2006.

REFERENCES

1. Gerstein M, Jansen R, Johnson T, Tsai J, Krebs W. Studying Macromolecular Motions in a Database Framework: from Structure to Sequence. Rigidity Theory Appl 1999:401–442.
2. Krebs W. The database of macromolecular motions: a standardized system for analyzing and visualizing macromolecular motions in a database framework. Dissertation. New Haven: Yale University.
3. Shatsky M, Nussinov R, Wolfson HJ. Flexible protein alignment and hinge detection. Proteins 2002;48:242–256.
4. Janin J, Wodak SJ. Structural domains in proteins and their role in the dynamics of protein function. Prog Biophys Mol Biol 1983; 42:21–78.
5. Thorpe MF, Jacobs DJ, Chubynsky MV, Phillips JC. Self-organization in network glasses. J Non-Cryst Solids 2000;266–269:859–866.
6. Jacobs DJ, Rader A, Kuhn LA, Thorpe MF. Protein flexibility predictions using graph theory. Proteins 2001;44:150–165.
7. Thorpe MF, et al. Protein flexibility and dynamics using constraint theory. J Mol Graph Model 2001;19:60–69.
8. Hespenheide BM, et al. Identifying protein folding cores from the evolution of flexible regions during unfolding. J Mol Graph Model 2002;21:195–207.
9. Rader AJ, et al. Protein unfolding: rigidity lost. Proc Natl Acad Sci USA 2002;99:3540–3545.
10. Wells S, et al. Constrained geometric simulation of diffusive motion in proteins. Phys Biol 2005;24:S127– S136.
11. Flores S, et al. The database of macromolecular motions: new features added at the decade mark. Nucleic Acids Res 2006;34 (Database issue):D296–D301.
12. 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.
13. Kundu S, Sorensen DC, Phillips GN, Jr. Automatic domain decomposition of proteins by a Gaussian Network Model. Proteins 2004; 57:725–733.
14. Dumontier M, et al. Armadillo: domain boundary prediction by amino acid composition. J Mol Biol 2005;350:1061–1073.
15. Flores S, et al. Hinge Atlas: relating sequence features to sites of structural flexibility. BMC Bioinformatics 2007:167.
16. Nagarajan N, Yona G. Automatic prediction of protein domains from sequence information using a hybrid learning system. Bioinformatics 2004;20:1335–1360.
17. Marsden RL, McGuffin LJ, Jones DT. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. Protein Sci 2002;11:2814–2824.
18. Schlessinger A, Rost B. Protein flexibility and rigidity predicted from sequence. Proteins 2005;61:115–126.
19. Painter J, Merritt EA. A molecular viewer for the analysis of TLS rigid-body motion in macromolecules. Acta Crystallogr D Biol Crystallogr 2005;61 (Part 4):465–471.
20. Flores S, Gerstein M. FlexOracle: predicting hinges by identification of stable domains. BMC Bioinformatics 2007;8:215.
21. Yang LW, Bahar I. Coupling between catalytic site and collective dynamics: a requirement for mechanochemical activity of enzymes. Structure 2005;13:893–904.
22. Nicolay S, Sanejouand YH. Functional modes of proteins are among the most robust. Phys Rev Lett 2006;96:078104.
23. Alexandrov V, et al. Normal modes for predicting protein motions: a comprehensive database assessment and associated Web tool. Protein Sci 2005;14:633–643.
24. Krebs WG, et al. Normal mode analysis of macromolecular motions in a database framework: developing mode concentration as a useful classifying statistic. Proteins 2002;48:682–695.
25. Chennubhotla C, et al. Elastic network models for understanding biomolecular machinery: from enzymes to supramolecular assemblies. Phys Biol 2005;2:S173–S180.
26. Alexandrov V, Gerstein M. Using 3D Hidden Markov Models that explicitly represent spatial coordinates to model and compare protein structures. BMC Bioinformatics 2004;5:2.
27. Yang LW, et al. iGNM: a database of protein functional motions based on Gaussian Network Model. Bioinformatics 2005;21:2978–2987.
28. Tirion MM. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. Phys Rev Lett 1996;77:1905–1908.
29. Haliloglu T, Bahar I, Erman B. Gaussian dynamics of folded proteins. Phys Rev Lett 1997;79:4.
30. Holm L, Sander C. Parser for protein folding units. Proteins 1994; 19:256–268.
31. Maxwell J. On the calculation of the equilibrium and stiffness of frames. Philos Mag 1864;27:294–299.
32. Tay T-S, whiteley W. Recent advances in generic rigidity of structures. Struct Topol 1985;9:31–38.
33. Jacobs DJ, Bruce H. An algorithm for two-dimensional rigidity percolation: The pebble game. J Comput Phys 1997;137:346–365.
34. Lindahl E, Hess B, van der Spoel D. GROMACS 3.0: A package for molecular simulation and trajectory analysis. J Mol Model 2001;7: 306–317.
35. Berendsen HJ, van der Spoel D, van Drunen R. GROMACS: A message-passing parallel molecular dynamics implementation. Comput Phys Commun 1995;91:43–56.
36. Gohlke H, Kuhn LA, Case DA. Change in protein flexibility upon complex formation: analysis of Ras-Raf using molecular dynamics and a molecular framework approach. Proteins 2004;56:322–37.
37. Painter J, Merritt EA. Optimal description of a protein structure in terms of multiple groups undergoing TLS motion. Acta Crystallogr D Biol Crystallogr 2006;62 (Part 4):439–450.
38. Schomaker V, Trueblood KN. On the rigid-body motion of molecules in crystals. Acta Crystallogr B 1968;24:63–76.
39. Winn MD, Isupov MN, Murshudov GN. Use of TLS parameters to model anisotropic displacements in macromolecular refinement. Acta Crystallogr D Biol Crystallogr 2001;57 (Part 1):122–133.
40. Sedgwick B, et al. Functional domains and methyl acceptor sites of the *Escherichia coli* ada protein. J Biol Chem 1988;263:4430–4433.
41. Jones S, et al. Domain assignment for protein structures using a consensus approach: characterization and analysis. Protein Sci 1998; 7:233–242.

42. Drapper N, Smith H. Applied regression analysis, 2nd ed. Wiley Series in Probability and Mathematical Statistics. Chichester, GB: Wiley; 1981.
43. Schapire R. The boosting approach to machine learning: an overview. MSRI Workshop on Nonlinear Estimation and Classification, 2002.
44. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. The Annals of Statistics 2000;38:337–374.
45. Culp M, Johnson K, Michailidis G. ada: an R package for stochastic boosting. Journal of Statistical Software 2006;17(2).
46. Chawla NV, et al. J Artif Intelligence Res 2002;16:341–378.
47. Pang A, et al. Interdomain dynamics and ligand binding: molecular dynamics simulations of glutamine binding protein. FEBS Lett 2003; 550:168–174.
48. Hsiao CD, et al. The crystal structure of glutamine-binding protein from *Escherichia coli*. J Mol Biol 1996;262:225–242.
49. Laskowski RA, et al. PROCHECK: A program to check the stereochemical quality of protein structures. J Appl Cryst 1993;26:283–291.
50. Davidsson L, et al. Influence of lactoferrin on iron absorption from human milk in infants. Pediatr Res 1994;35:117–124.
51. Gerstein M, et al. Domain closure in lactoferrin. Two hinges produce a see-saw motion between alternative close-packed interfaces. J Mol Biol 1993;234:357–372.
52. Norris GE, Anderson BF, Baker EN. Molecular replacement solution of the structure of apolactoferrin, a protein displaying large-scale conformational change. Acta Crystallogr B 1991;47 (Part 6):998–1004.
53. 2007 10 November 2007 [cited; Available from: http://en.wikipedia.org/wiki/Overfitting].