Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Molecular Graphics and Modelling 22 (2004) 183–193
Journal of Molecular Graphics and Modelling
www.elsevier.com/locate/JMGM

Normal mode analysis of proteins: a comparison of rigid cluster modes with C_α coarse graining

Adam D. Schuyler, Gregory S. Chirikjian*

Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218, USA

Received 5 March 2003; received in revised form 22 June 2003; accepted 22 June 2003

Abstract

The ability to infer dynamic motions from an equilibrium (static) conformation of a protein can be essential in establishing structure–function relationships. In particular, the low-frequency motions are of functional interest because statistical mechanics predicts these motions will have the largest amplitudes. In this paper, we address the computational cost of normal mode analysis (NMA) applied to a C_α-based elastic network model (C_α-NMA) and present a new coarse-grained rigid-body-based analysis (cluster-NMA). This new method represents a protein as a collection of rigid bodies interconnected with harmonic potentials. This representation produces reduced degree-of-freedom (DOF) equations of motion (EOMs) which, even in the case of large structures (10^{3+} residues), enables the computation of normal modes to be done on a desktop PC. We present the complete theory and analysis of cluster-NMA and also include its application to a variety of structures. The results of the new method are compared with C_α-NMA and it is shown that cluster-NMA produces very good approximations to the lowest modes at a fraction of the computational cost.

Keywords: Normal mode analysis; Protein mechanics; Rigid-body motion; Coarse grain; Cluster-NMA; Conformational transition

1. Introduction

The search for inherent links between protein structure and function is a driving force behind the development of accurate and computationally efficient methods for describing the complex motions attainable by protein structures. X-ray crystallography can provide snapshots of protein structures in (near) equilibrium conformations.[1] Other experimental methods such as fluorescent resonance energy transfer (FRET) and nuclear magnetic resonance (NMR) can provide partial information about large-amplitude protein motions [2–4]. Dynamical simulations of protein structures can provide crucial insights into their function which are not easily obtained experimentally. It has been observed that the low-frequency, large-amplitude motions are most closely related to protein function [5–8] whereas the high-frequency localized vibrations may be more involved in signal transmission and other internal processes [9]. Computational models (normal mode analysis (NMA) in particular) enable one to derive these desired dynamic motions from the static conformations obtained from crystallography and are thus an essential tool in gaining insight into the structure–function relationship.

Molecular dynamics (MD) simulations rely on atomic details to predict the evolution of conformations of protein structures based on the interactions between all pairs of atoms. These simulations can be computationally prohibitive due to the high number of degrees-of-freedom (DOFs) required to capture motions of large structures and the complicated force calculations required at each iteration.

As a first level of simplification, consider the basic structure of any protein. A protein is comprised of a polypeptide backbone with amino acid residues extending from the alpha-carbon (C_α) of each peptide unit. Since the peptide-bonded backbone remains connected in all conformations (short of denaturing the protein), it is often useful to focus on the backbone structure knowing that each side chain must closely follow its corresponding C_α. Bahar et al. [10] present a scalar model, called the Gaussian network model (GNM), which produces magnitudes of individual residue displacements consistent with experimentally derived quantities including X-ray crystallographic temperature factors [7], hydrogen exchange free energies [11], and the order parameters from NMR-relaxation measurements [12]. While these results validate the use of simple

* Tel.: +410-516-7127; fax: +410-516-7254.
E-mail address: gregc@hu.edu (G.S. Chirikjian).
[1] Due to thermal fluctuations a protein structure adopts a range of conformations about its thermal equilibrium [1].

1093-3263/$ – see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S1093-3263(03)00158-X

elastic networks, they do not produce displacement directions. Atilgan et al. [13] present the anisotropic network model (ANM) which builds on the GNM by including parameters for displacement directions. Similarly, Kim et al. [14,15] use C_α-NMA, in which the interactions between residues in contact are modelled with harmonic potentials, to produce three-dimensional displacements. While these methods are much faster than all-atom simulations, they are still computationally expensive for very large structures.

C_α-NMA, as mentioned above, is one of the highest resolution coarse-grained models (one C_α per grain). It uses the Cartesian displacements of each C_α to define the conformational displacement relative to the initial conformation [16]. Coarser-grained models have been employed to capture large-amplitude motions. For example, coarse-graining methods have been employed by [17], in which the full protein structure is projected onto a reduced DOF subspace. The hybrid method MBO(N)D, as presented in [18], makes use of varying grain sizes to achieve desired levels of resolution according to the mobility and functional interest of each region within the structure.

Hinsen [5] uses a Fourier basis to capture a uniform vector field of displacements. This reduced degree-of-freedom model effectively captures the lowest modes, but has a number of limitations which are inherent to a Fourier basis: it is not well suited for capturing translational motion, periodicity of the basis set must be accounted for, displacements given in the basis coordinates do not have physical meaning.

Central to every modelling method is the choice of parameterization. In general, higher DOF parameterizations allow more complex motions to be captured (at a significant computational cost), while lower DOF parameterizations can impose unrealistic conformational constraints. The choice of parameterization allows the user to attain the desired combination of computational performance and motion resolution. This trade-off can be adjusted within a given structure so that regions of interest can be modelled with higher resolution than other regions of less importance. In this paper, we present a low DOF parameterization that produces low-frequency motions consistent with C_α-NMA, which in turn has shown strong agreement with all-atom NMA and MD simulations [5,10,19].

An n residue structure requires $3n$ parameters for full resolution C_α-NMA. We refer to this as the standard parameterization,[2] as it serves as our basis for comparison. This results in a computational complexity of $\mathcal{O}(n^3)$.[3] However, as mentioned above, the modes of interest are the low-frequency, large-amplitude motions and not the high-frequency localized vibrations (i.e. one is typically not interested in all $3n$ modes).[4] We bypass these issues with clustering algorithms to identify subsets of residues that form rigid clusters and thus move as rigid units under the modes of interest.

When multiple conformations are known, as in the case of lactoferrin (PDB: 1LFG and 1LFH), the structure can be clustered by identifying sets of residues that experience minimal RMS deviation (after optimal alignment of each candidate cluster). For all structures there are algorithms such as the pebble game [21] that count DOF constraints in a network of contacts. Proteins can also be clustered by secondary structure elements. In all cases, clustering effectively filters out the high-frequency modes and enables one to use a low DOF parameterization to more efficiently calculate the global modes.

The rest of this paper is organized as follows. In Section 2.1, the C_α elastic network model is reviewed. In Section 2.2, the central points of clustering algorithms are discussed and cluster notation is introduced. In Section 2.3, the parameters necessary to capture the motions of this system of rigid bodies are defined. In Sections 2.4 and 2.5, the stiffness and mass matrices are obtained from the quadratic expressions for the potential and kinetic energies of the system. In Section 2.6, the mode shapes are extracted from the equation of motion (EOM) and projected onto the structure. This process requires a change of coordinates, the Gram–Schmidt orthonormalization process, and a low mode "unmixing" algorithm. In Section 3, cluster-NMA is applied to a variety of structures. Computational performance and mode accuracy are analyzed. In Section 4, a summary analysis is given.

2. Method

2.1. Review of C_α-NMA

Since cluster-NMA will be compared to C_α-NMA, the C_α model [13,14], is briefly reviewed here for completeness. Structures are represented as a system of point masses located at each C_α position with a network of connecting springs. Conformational changes are viewed as displacements of each C_α in the structure. The vector of generalized coordinates is $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_1^{\mathrm{T}}, \ldots, \boldsymbol{\sigma}_n^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{\sigma}_i \in \mathbb{R}^3$ is the displacement of residue i and $\mathbb{R}^d$ denotes the d-dimensional space of real valued vectors.

The $3n \times 3n$ mass matrix and stiffness matrix in this model are defined by constructing an $n \times n$ array of 3×3 matrices. The mass matrix sub-blocks are of the form

$$[M_{\mathrm{s}}]_{i,j} = \begin{cases} m_i \mathbb{I}_3, & \text{for } i = j \\ 0_3, & \text{for } i \neq j \end{cases} \tag{1}$$

[2] Another common parameterization uses the internal torsion angles (ϕ, ψ). This requires $\mathcal{O}(n)$ parameters as well.

[3] There are other methods besides Gaussian elimination that are mentioned in [20] that can reduce the exponent as low as 2.376. However, these methods typically require a complicated initialization and can have numerical stability problems. As a result NMA (a matrix multiplication/inversion dependent calculation) is commonly considered to have complexity $\mathcal{O}(n^3)$.

[4] There are iterative methods for determining a partial set of eigenpairs, but these often require initial guesses and can often have numerical stability problems. For these reasons, we still classify the eigenproblem as $\mathcal{O}(n^3)$.

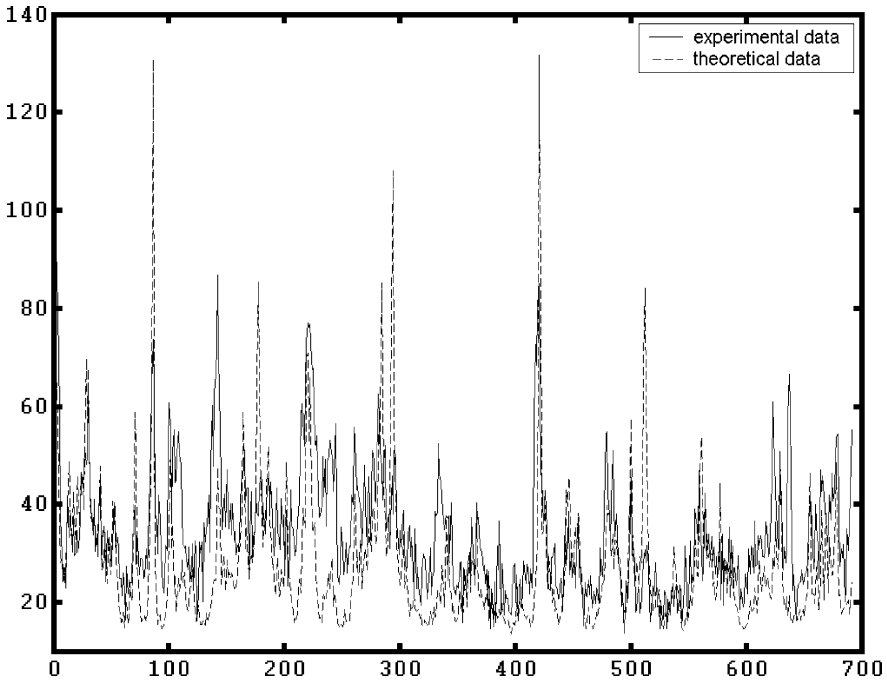

Fig. 1. Comparison between experimentally and theoretically derived temperature factor data for lactoferrin (PDB: 1LFH).

where m_i is the mass of residue i, $\mathbb{I}_3$ is the 3×3 identity matrix, and 0_3 is the 3×3 zero matrix. The stiffness matrix sub-blocks are defined by

$$[K_s]_{i,j} = \begin{cases} -k_{i,j} \dfrac{(\boldsymbol{r}_i - \boldsymbol{r}_j)(\boldsymbol{r}_i - \boldsymbol{r}_j)^T}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2}, & \text{for } i \neq j \\ -\sum_{r=1}^{i-1} K_{r,i} - \sum_{c=i+1}^{n} K_{i,c}, & \text{for } i = j \end{cases} \quad (2)$$

where $\boldsymbol{r}_i$ is the Cartesian location of residue i and the inter-residue interactions are determined by the simple[5] spring constant expression

$$k_{i,j} = \begin{cases} 1 & \|\boldsymbol{r}_i - \boldsymbol{r}_j\| \leq r \\ 0 & \|\boldsymbol{r}_i - \boldsymbol{r}_j\| > r \end{cases} \quad (3)$$

These interactions can be further modified by enforcing a maximum contact number. This constraint is achieved by iteratively disconnecting the furthest neighbor from the residue with the most contacts until no residues have more than the allowed maximum. The resulting EOM is

$$M_s \ddot{\boldsymbol{\sigma}} + K_s \boldsymbol{\sigma} = \mathbf{0} \quad (4)$$

The C_α-NMA model can be used to produce theoretical temperature factor data, which compares favorably with the experimentally derived quantity. For example, Fig. 1 shows such calculations for lactoferrin. The two plots are very similar with the largest differences occurring near peak values where the theoretical model occasionally over-estimates. The data is optimally aligned by choosing a single scaling parameter that corresponds to the single spring constant in the elastic network. With this parameter, we minimize the least-square error between the experimental data and the theoretically derived temperature factor. This analysis follows as given in [13].

2.2. Clustering and system setup

A clustering algorithm is primarily used to reduce the number of DOFs in the protein structure to speed up the modal analysis. Since, as demonstrated by the pebble game [21], there exists such an algorithm that performs in the worst case in $\mathcal{O}(n^2)$ operations (and more typically in $\mathcal{O}(n)$), it is not necessary to concern ourselves with optimizing a clustering algorithm. In this paper, a helix-based clustering algorithm is used. Energetically favorable helix geometries tend to hold shape, especially during the low-frequency motions of interest. The helix boundaries are used to form an initial partition on the structure. A target cluster size is used to evenly (as possible) partition up the remaining residues and any helices that are larger than the target size. The end goal of this method is to achieve a uniform cluster size specified by the target size and have the helix boundaries aligned with the cluster interfaces. This straightforward $\mathcal{O}(n)$ method is used for the examples given in this paper.

The following notation accommodates any choice of clustering algorithm. Let such an algorithm produce a set of N rigid clusters

$$C = \{C_1, \ldots, C_N\} \quad (5)$$

[5] In place of this discrete function, one can use continuous functions that produce a more Lennard–Jones-like potential. Regardless of what function is used, this formalism remains valid for small motions around equilibrium. We will use the binary function for simplicity of presentation.

where all residues are in exactly one cluster (i.e. the partition is complete and disjoint). The time varying global frame positions of the residues in cluster i are denoted as

$$C_i(t) = \{\boldsymbol{r}_{i,1}(t), \ldots, \boldsymbol{r}_{i,N(i)}(t)\} \tag{6}$$

where $\boldsymbol{r}_{i,\alpha} \in \mathbb{R}^3$ is used to represent the Cartesian position of residue α in cluster i with the range of cluster indices given by $i \in [1, \ldots, N]$ and the range of residue indices in cluster i given by $\alpha \in [1, \ldots, N(i)]$. In subsequent sections, position and orientation of these clusters are calculated. This brings us to the possibility of a special case: the trivial cluster. Any cluster that does not have at least three non-collinear residues is not uniquely orientable. To handle this situation, we choose to break all trivial clusters into their constituent residues and reassign each to the cluster that has the most residues in contact with the candidate residue.

2.3. Defining parameters

Coordinates of protein structures are obtained from the protein data bank [22] and define the reference conformation. The translational and orientational motions of clusters are measured with respect to this conformation. Any conformation is thus parameterized by $6N$ DOFs (i.e. 3 DOFs for translational motion and 3 DOFs for orientational motion for each of the N clusters).

2.3.1. Translational displacement

The translational movement of each cluster is monitored by the motion of its center of mass

$$\boldsymbol{x}_i(t) = \frac{1}{\bar{m}_i} \sum_{\alpha=1}^{N(i)} m_{i,\alpha} \boldsymbol{r}_{i,\alpha}(t) \tag{7}$$

where $m_{i,\alpha}$ is the mass of residue α of cluster i and $\bar{m}_i$ is the mass of cluster i. The translational displacement parameter is thus defined as

$$\boldsymbol{\chi}_i(t) = \boldsymbol{x}_i(t) - \boldsymbol{x}_i(0) \tag{8}$$

which gives the displacement of the center of mass of $C_i(t)$, where $t = 0$ corresponds to the crystal structure.

2.3.2. Orientational displacement

As with the translational displacement, orientational displacement is measured relative to the initial conformation. This is equivalent to assigning an identity reference frame to each cluster (i.e. it is considered aligned with the base frame at time 0) and then monitoring the displacement from this orientation. To parameterize this motion, let cluster i have an orientation of $R_i(t)$, which is an element of the set of 3×3 rotation matrices. The orientational displacement parameter, $\boldsymbol{\gamma}_i(t)$, is defined by its relation to the rotation matrix, $R_i(t)$, in the expression[6]

$$\begin{aligned} R_i(t) &= \mathrm{e}^{J(\boldsymbol{\gamma}_i(t))} \\ R_i(t) &= \mathbb{I}_3 + \left(\frac{\sin(\|\boldsymbol{\gamma}_i(t)\|)}{\|\boldsymbol{\gamma}_i(t)\|}\right)[J(\boldsymbol{\gamma}_i(t))] \\ &\quad + \left(\frac{1 - \cos(\|\boldsymbol{\gamma}_i(t)\|)}{\|\boldsymbol{\gamma}_i(t)\|^2}\right)[J(\boldsymbol{\gamma}_i(t))]^2 \\ R_i(t) &\triangleq R(\boldsymbol{\gamma}_i(t)) \end{aligned} \tag{10}$$

where $\|\boldsymbol{\gamma}_i(t)\|$ is the usual vector norm.

2.3.3. Generalized coordinate

From the previous two sections we can now define our cluster pose parameter as

$$\boldsymbol{\delta}_i(t) = \begin{pmatrix} \boldsymbol{\chi}_i(t) \\ \boldsymbol{\gamma}_i(t) \end{pmatrix} \in \mathbb{R}^6 \tag{11}$$

and our system generalized coordinate as

$$\boldsymbol{\delta}(t) = \begin{pmatrix} \boldsymbol{\delta}_1(t) \\ \vdots \\ \boldsymbol{\delta}_N(t) \end{pmatrix} \in \mathbb{R}^{6N} \tag{12}$$

2.4. Derivation of stiffness matrix

In Section 2.4.1, the location of an arbitrary residue under an arbitrary cluster motion is determined. This result provides the locations of all spring endpoints. In Section 2.4.2, the displacement across an arbitrary spring is determined. From this quantity, the potential energy of the system is directly calculated. In Section 2.4.3, algebraic manipulations are performed to achieve a representation where the potential energy equation is quadratic in $\boldsymbol{\delta}$.

2.4.1. Arbitrary residue location

In C_i the springs that branch out to other clusters (which are the springs that contribute to the potential energy of the system) are numbered. This produces the set of spring endpoints in global frame coordinates

$$S_i(t) = \{\boldsymbol{s}_{i,1}(t), \ldots, \boldsymbol{s}_{i,M(i)}(t)\} \tag{13}$$

where $M(i)$ is the number of springs that have exactly one endpoint in C_i.

Consider the position of an arbitrary spring endpoint, $\boldsymbol{s}_{i,\alpha}(t)$, under an arbitrary cluster motion, $\boldsymbol{\delta}_i(t)$. Its position

[6] This equation uses the skew-symmetric matrix function, J, to map elements of $\mathbb{R}^3$ onto elements of SK(3), which is the set of 3×3 skew-symmetric matrices. This function is given by

$$J(\boldsymbol{\gamma}_i) = \begin{bmatrix} 0 & -(\boldsymbol{\gamma}_i)_z & (\boldsymbol{\gamma}_i)_y \\ (\boldsymbol{\gamma}_i)_z & 0 & -(\boldsymbol{\gamma}_i)_x \\ -(\boldsymbol{\gamma}_i)_y & (\boldsymbol{\gamma}_i)_x & 0 \end{bmatrix}, \quad \text{where}$$

$$\boldsymbol{\gamma}_i = \begin{pmatrix} (\boldsymbol{\gamma}_i)_x \\ (\boldsymbol{\gamma}_i)_y \\ (\boldsymbol{\gamma}_i)_z \end{pmatrix} \tag{9}$$

The parameter $\boldsymbol{\gamma}_i(t)$ is the Rodrigues vector. Its magnitude is the corresponding angle of rotation of $R_i(t)$ and its direction is the axis of rotation.

is given by

$$s_{i,\alpha}(t) = \underbrace{[R(\gamma_i(t))](s_{i,\alpha}(0) - x_i(0)) + x_i(0)}_{\text{rotate residue about its cluster's center of mass}} + \underbrace{\chi_i(t)}_{\text{translate residue}} \tag{14}$$

In Eq. (14), the cluster is shifted so that its center of mass moves to the global frame's origin, then the rotation is applied, then the cluster's center of mass is shifted back to its original position, and finally the cluster is translated by an amount $\chi_i(t)$. The order of these operations is determined by how the pose parameters have been defined. Fig. 2 is a 2D representation of the quantities involved.

Since the oscillatory motions of a system about its equilibrium conformation are being modelled, small motions are assumed. The first-order approximation of Rodrigues' formula in Eq. (10) is used to obtain

$$\begin{aligned} s_{i,\alpha}(t) &\approx [\mathbb{I}_3 + J(\gamma_i(t))](s_{i,\alpha}(0) - x_i(0)) + x_i(0) + \chi_i(t) \\ &= s_{i,\alpha}(0) + [\mathbb{I}_3, -J(s_{i,\alpha}(0) - x_i(0))] \begin{pmatrix} \chi_i(t) \\ \gamma_i(t) \end{pmatrix} \\ &\triangleq s_{i,\alpha}(0) + [Q_{i,\alpha}]\delta_i(t) \end{aligned} \tag{15}$$

2.4.2. Displacement across arbitrary spring

Consider any spring in the network and, by Eq. (15), let its endpoints be given in global frame coordinates by

$$s_{i,\alpha}(t) = s_{i,\alpha}(0) + [Q_{i,\alpha}]\delta_i(t) \tag{16}$$

$$s_{j,\beta}(t) = s_{j,\beta}(0) + [Q_{j,\beta}]\delta_j(t) \tag{17}$$

The square of the change in length of this spring from its value at equilibrium is

$$\begin{aligned} d^2_{i,\alpha,j,\beta}(t) &\triangleq (\|s_{i,\alpha}(t) - s_{j,\beta}(t)\| - \|s_{i,\alpha}(0) - s_{j,\beta}(0)\|)^2 \\ &= \left(\| \underbrace{(s_{i,\alpha}(0) - s_{j,\beta}(0))}_{a} + \underbrace{([Q_{i,\alpha}]\delta_i(t) - [Q_{j,\beta}]\delta_j(t))}_{b} \| - \| \underbrace{s_{i,\alpha}(0) - s_{j,\beta}(0)}_{a} \| \right)^2 \\ &= \|a + b\|^2 + \|a\|^2 - 2\|a + b\|\|a\| \approx (2\|a\|^2 + 2ab + \|b\|^2) - 2\left(\|a\| + \frac{ab}{\|a\|} + \frac{b^{\mathrm{T}}[\mathbb{I} - Y(a)]b}{2\|a\|} \right)\|a\| \\ &= b^{\mathrm{T}}[Y(a)]b \end{aligned} \tag{18}$$

which uses the definition

$$Y(a) = \frac{aa^{\mathrm{T}}}{\|a\|^2} \tag{19}$$

By defining the quantities

$$\Delta_{i,j}(t) = \begin{pmatrix} \delta_i(t) \\ \delta_j(t) \end{pmatrix} \in \mathbb{R}^{12} \tag{20}$$

and

$$Q_{i,\alpha,j,\beta} = [Q_{i,\alpha} - Q_{j,\beta}] \in \mathbb{R}^{3\times 12} \tag{21}$$

the expression

$$[Q_{i,\alpha}]\delta_i(t) - [Q_{j,\beta}]\delta_j(t) = Q_{i,\alpha,j,\beta}\Delta_{i,j}(t) \tag{22}$$

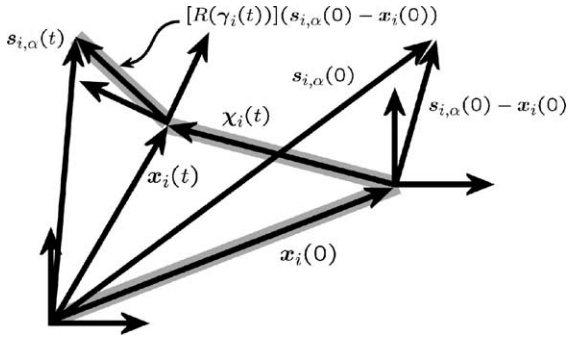

Fig. 2. This 2D representation shows the relationship between all quantities used to describe the position of $s_{i,\alpha}(t)$. Eq. (14) can be derived from this figure by summing all vector quantities along the shaded path.

is obtained. The squared spring displacement is rewritten as

$$\begin{aligned} &d^2_{i,\alpha,j,\beta}(t) \\ &\quad = (Q_{i,\alpha,j,\beta}\Delta_{i,j}(t))^{\mathrm{T}} \underbrace{[Y(s_{i,\alpha}(0) - s_{j,\beta}(0))]}_{Y_{i,\alpha,j,\beta}} (Q_{i,\alpha,j,\beta}\Delta_{i,j}(t)) \\ &\quad = \Delta^{\mathrm{T}}_{i,j}(t) \underbrace{[Q^{\mathrm{T}}_{i,\alpha,j,\beta} Y_{i,\alpha,j,\beta} Q_{i,\alpha,j,\beta}]}_{S_{i,\alpha,j,\beta}} \Delta_{i,j}(t) \end{aligned} \tag{23}$$

The matrix S, which is related to the desired stiffness matrix, K, is symmetric:

$$S_{i,\alpha,j,\beta} = \left[\begin{array}{c|c|c|c} Y_{i,\alpha,j,\beta} & [J_{i,\alpha}Y_{i,\alpha,j,\beta}]^{\mathrm{T}} & -[Y_{i,\alpha,j,\beta}]^{\mathrm{T}} & -[J_{j,\beta}Y_{i,\alpha,j,\beta}]^{\mathrm{T}} \\ J_{i,\alpha}Y_{i,\alpha,j,\beta} & -J_{i,\alpha}Y_{i,\alpha,j,\beta}J_{i,\alpha} & [Y_{i,\alpha,j,\beta}J_{i,\alpha}]^{\mathrm{T}} & [J_{j,\beta}Y_{i,\alpha,j,\beta}J_{i,\alpha}]^{\mathrm{T}} \\ -Y_{i,\alpha,j,\beta} & Y_{i,\alpha,j,\beta}J_{i,\alpha} & Y_{i,\alpha,j,\beta} & [J_{j,\beta}Y_{i,\alpha,j,\beta}]^{\mathrm{T}} \\ -J_{j,\beta}Y_{i,\alpha,j,\beta} & -J_{j,\beta}Y_{i,\alpha,j,\beta}J_{i,\alpha} & J_{j,\beta}Y_{i,\alpha,j,\beta} & -J_{j,\beta}Y_{i,\alpha,j,\beta}J_{j,\beta} \end{array}\right] \tag{24}$$

2.4.3. Extraction of stiffness matrix

The system's potential energy is determined by summing over all spring contributions. This is achieved by considering all cluster pairs followed by all springs connecting each pair. The resulting expression is

$$V(t) = \frac{1}{2}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \Delta_{i,j}(t)^{\mathrm{T}} \underbrace{\left[\sum_{\alpha=1}^{M(i)}\sum_{\beta=1}^{M(j)} k_{i,\alpha,j,\beta}S_{i,\alpha,j,\beta}\right]}_{\kappa_{i,j}} \Delta_{i,j}(t) \tag{25}$$

where the spring constants are defined by

$$k_{i,\alpha,j,\beta} = \begin{cases} 1 & \|\boldsymbol{s}_{i,\alpha}(0) - \boldsymbol{s}_{j,\beta}(0)\| \leq r \\ 0 & \|\boldsymbol{s}_{i,\alpha}(0) - \boldsymbol{s}_{j,\beta}(0)\| > r \end{cases} \tag{26}$$

A maximum contact number can also be imposed.

$\boldsymbol{\Delta}_{i,j}$ is a stacked vector of two cluster displacements ($\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$). The summation indices in Eq. (25) allow for multiple appearances of each. We seek the matrix K that produces the equation

$$V(t) = \tfrac{1}{2}\boldsymbol{\delta}(t)^{\mathrm{T}} K \boldsymbol{\delta}(t) \tag{27}$$

Eq. (24) shows the symmetry of $S_{i,j,\alpha,\beta}$ and Eq. (25) shows that $\kappa_{i,j}$ is formed by summing a collection of symmetric matrices. This allows the $\mathbb{R}^{6\times 6}$ sub-matrices of $\kappa_{i,j}$ to be defined as

$$\kappa_{i,j} = \begin{bmatrix} A_{i,j} & B_{i,j} \\ B_{i,j}^{\mathrm{T}} & C_{i,j} \end{bmatrix} \tag{28}$$

From the potential energy expression it can be determined where the sub-blocks of $\kappa_{i,j}$ belong in K. The expansion

$$\boldsymbol{\Delta}_{i,j}^{\mathrm{T}} \boldsymbol{\kappa}_{i,j} \boldsymbol{\Delta}_{i,j} = \boldsymbol{\delta}_i^{\mathrm{T}} A_{i,j} \boldsymbol{\delta}_i + \boldsymbol{\delta}_j^{\mathrm{T}} C_{i,j} \boldsymbol{\delta}_j + 2\boldsymbol{\delta}_i^{\mathrm{T}} B_{i,j} \boldsymbol{\delta}_j \tag{29}$$

leads to the expression

$$K_{r,c} = \begin{cases} \left[\sum_{j=r+1}^{N} A_{r,j}\right] + \left[\sum_{i=1}^{c-1} C_{i,c}\right], & r = c \\ [B_{r,c}], & r \neq c \end{cases} \tag{30}$$

where K is defined as

$$K = \begin{bmatrix} K_{1,1} & \cdots & K_{1,N} \\ \vdots & \ddots & \vdots \\ K_{N,1} & \cdots & K_{N,N} \end{bmatrix} \in \mathbb{R}^{6N\times 6N} \tag{31}$$

2.5. Derivation of the mass matrix

2.5.1. General formulation

The mass matrix is derived from the quadratic expression for the kinetic energy. The kinetic energy of a collection of rigid clusters can be written as the sum of contributions from translational and rotational kinetic energies. The desired expression is of the form

$$T(t) = T_{\mathrm{trans}}(t) + T_{\mathrm{rot}}(t) = \tfrac{1}{2}\dot{\boldsymbol{\delta}}(t)^{\mathrm{T}} M \dot{\boldsymbol{\delta}}(t) \tag{32}$$

To solve for M in the above equation, consider the kinetic energy of a single cluster. In block form, the expression becomes

$$\begin{aligned} T_i(t) &= \frac{1}{2}(\dot{\boldsymbol{\chi}}_i(t)^{\mathrm{T}}, \dot{\boldsymbol{\gamma}}_i(t)^{\mathrm{T}}) \underbrace{\begin{bmatrix} M_{i,\mathrm{trans}} & 0_3 \\ 0_3 & M_{i,\mathrm{rot}} \end{bmatrix}}_{M_i} \begin{pmatrix} \dot{\boldsymbol{\chi}}_i \\ \dot{\boldsymbol{\gamma}}_i \end{pmatrix} \\ &= \tfrac{1}{2}\dot{\boldsymbol{\chi}}_i(t)^{\mathrm{T}} M_{i,\mathrm{trans}} \dot{\boldsymbol{\chi}}_i(t) + \tfrac{1}{2}\dot{\boldsymbol{\gamma}}_i(t)^{\mathrm{T}} M_{i,\mathrm{rot}} \dot{\boldsymbol{\gamma}}_i(t) \end{aligned} \tag{33}$$

The sub-blocks of the mass matrix are determined by solving the following equations

$$\tfrac{1}{2}\bar{m}_i \|\dot{\boldsymbol{x}}_i(t)\|^2 = \tfrac{1}{2}\dot{\boldsymbol{\chi}}_i(t)^{\mathrm{T}} M_{i,\mathrm{trans}} \dot{\boldsymbol{\chi}}_i(t) \tag{34}$$

$$\tfrac{1}{2}\boldsymbol{\omega}_i(t)^{\mathrm{T}} I_i(t) \boldsymbol{\omega}_i(t) = \tfrac{1}{2}\dot{\boldsymbol{\gamma}}_i(t)^{\mathrm{T}} M_{i,\mathrm{rot}} \dot{\boldsymbol{\gamma}}_i(t) \tag{35}$$

where $\boldsymbol{\omega}_i$ is the angular velocity of cluster i. The moment of inertia matrix of cluster i with respect to the center of mass of cluster i is defined by

$$I_i = \sum_{\alpha=1}^{N(i)} m_i (\hat{\boldsymbol{r}}_{i,\alpha}^{\mathrm{T}} \hat{\boldsymbol{r}}_{i,\alpha} \mathbb{I}_3 - \hat{\boldsymbol{r}}_{i,\alpha} \hat{\boldsymbol{r}}_{i,\alpha}^{\mathrm{T}}) \tag{36}$$

where the residue coordinates are given with respect to their corresponding cluster's center of mass in the crystal structure as $\hat{\boldsymbol{r}}_{i,\alpha} = \boldsymbol{r}_{i,\alpha}(0) - \boldsymbol{x}_i(0)$.

2.5.2. Translational contribution

To solve Eq. (34) for $M_{i,\mathrm{trans}}$, recall that $\boldsymbol{\chi}_i(t) = \boldsymbol{x}_i(t) - \boldsymbol{x}_i(0)$ and thus $\dot{\boldsymbol{\chi}}_i(t) = \dot{\boldsymbol{x}}_i(t)$. This relation produces the first result

$$\begin{aligned} \frac{1}{2}\bar{m}_i \|\dot{\boldsymbol{x}}_i(t)\|^2 &= \frac{1}{2}\dot{\boldsymbol{\chi}}_i(t)^{\mathrm{T}} [\bar{m}_i \mathbb{I}_3] \dot{\boldsymbol{\chi}}_i(t) \\ \Rightarrow M_{i,\mathrm{trans}} &= \begin{bmatrix} \bar{m}_i & 0 & 0 \\ 0 & \bar{m}_i & 0 \\ 0 & 0 & \bar{m}_i \end{bmatrix} \end{aligned} \tag{37}$$

2.5.3. Rotational contribution

The rotational term is slightly more complicated. Since the generalized coordinate parameterizes orientation with $\boldsymbol{\gamma}_i$, we need to relate $\dot{\boldsymbol{\gamma}}_i$ to the angular velocity vector, $\boldsymbol{\omega}_i$, by using the "right Jacobian",[7] $\mathcal{J}_i$, in the expression

$$\boldsymbol{\omega}_i(t) = [\mathcal{J}_i(\boldsymbol{\gamma}_i(t))] \dot{\boldsymbol{\gamma}}_i(t) \tag{38}$$

where

$$\begin{aligned} \mathcal{J}_i(\boldsymbol{\gamma}_i(t)) = \mathbb{I}_3 &- \left(\frac{1 - \cos(\|\boldsymbol{\gamma}_i(t)\|)}{\|\boldsymbol{\gamma}_i(t)\|^2}\right) [J(\boldsymbol{\gamma}_i(t))] \\ &+ \left(\frac{\|\boldsymbol{\gamma}_i(t)\| - \sin(\|\boldsymbol{\gamma}_i(t)\|)}{\|\boldsymbol{\gamma}_i(t)\|^3}\right) [J(\boldsymbol{\gamma}_i(t))]^2 \end{aligned} \tag{39}$$

Based on the small motion approximation and the desire to have a constant valued mass matrix, the Jacobian is evaluated at the initial configuration. This gives the simple relation

$$\boldsymbol{\omega}_i(t) \approx [\mathcal{J}_i(\boldsymbol{\gamma}_i(0))] \dot{\boldsymbol{\gamma}}_i(t) = \dot{\boldsymbol{\gamma}}_i(t) \tag{40}$$

which can be used to write the rotational kinetic energy in the form

$$\tfrac{1}{2}\boldsymbol{\omega}_i(t)^{\mathrm{T}} I_i \boldsymbol{\omega}_i(t) \approx \tfrac{1}{2}\dot{\boldsymbol{\gamma}}_i(t)^{\mathrm{T}} I_i \dot{\boldsymbol{\gamma}}_i(t) \Rightarrow M_{i,\mathrm{rot}} \approx I_i \tag{41}$$

2.5.4. Final form

Assembling the sub-blocks of M as derived in the previous two sections yields the final expression for the mass matrix

[7] The right Jacobian relates $\boldsymbol{\omega}_i(t)$ and $\dot{\boldsymbol{\gamma}}_i(t)$ in body-fixed coordinates and the left Jacobian relates the quantities in space-fixed coordinates. For the right Jacobian, one uses I_i as defined in Eq. (36). For the left Jacobian, one uses $I_i'(t) = [R_i(t)][I_i][R_i(t)]^{\mathrm{T}}$ where $R_i(t)$ relates body and space frames. For small orientational motion, $R_i(t) \approx \mathbb{I}_3$ and $I_i'(t) \approx I_i(t)$. See [23] (p. 130) for derivation of both Jacobians.

$$M = \begin{bmatrix} M_i & 0_6 & \cdots & 0_6 \\ 0_6 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_6 \\ 0_6 & \cdots & 0_6 & M_N \end{bmatrix}, \quad \text{where}$$

$$M_i = \begin{bmatrix} \bar{m}_i \mathbb{I}_3 & 0_3 \\ 0_3 & I_i \end{bmatrix} \tag{42}$$

2.6. Mode shape extraction

The derivations of the previous sections produce the equation

$$M\ddot{\boldsymbol{\delta}} + K\boldsymbol{\delta} = \mathbf{0} \tag{43}$$

Statistical mechanics dictates that this is an equation whose harmonic solutions contribute to the equilibrium motions of the protein with amplitudes proportional to the inverse of their frequencies.

2.6.1. EOM solution

The following manipulations are performed to produce the mode shapes of Eq. (43).

1. Multiply on the left by[8] $M^{-1/2} \Rightarrow M^{1/2}\ddot{\boldsymbol{\delta}} + M^{-1/2} K\boldsymbol{\delta} = \mathbf{0}$
2. Define the coordinate: $\boldsymbol{y} = M^{1/2}\boldsymbol{\delta} \Rightarrow \ddot{\boldsymbol{y}} + \underbrace{M^{-1/2}KM^{-1/2}}_{\bar{S}=\bar{S}^{\mathrm{T}}} \boldsymbol{y} = \mathbf{0}$
3. Calculate the eigenpairs of $\bar{S}$ as $\bar{S}\bar{V} = \bar{V}\bar{D}$ (the columns of $\bar{V}$ are the eigenvectors of $\bar{S}$ and the diagonal matrix $\bar{D}$ has the corresponding eigenvalues).
4. The column vectors of $V = M^{-1/2}\bar{V}$ are the mode shapes transformed back into $\boldsymbol{\delta}$ coordinates. We refer to the kth mode as $(\boldsymbol{\delta})_k = [(\boldsymbol{\delta}_1)_k^{\mathrm{T}}, \ldots, (\boldsymbol{\delta}_N)_k^{\mathrm{T}}]^{\mathrm{T}}$ where $(\boldsymbol{\delta}_i)_k = [(\boldsymbol{\chi}_i)_k^{\mathrm{T}} (\boldsymbol{\gamma}_i)_k^{\mathrm{T}}]^{\mathrm{T}}$ is the motion of cluster i under mode k.
5. For each mode shape, $(\boldsymbol{\delta})_k$, project the cluster motions onto the protein structure using

$$\boldsymbol{r}_{i,\alpha}(t) = [R((\boldsymbol{\gamma}_i)_k(t))](\boldsymbol{r}_{i,\alpha}(0) - \boldsymbol{x}_i(0)) + \boldsymbol{x}_i(0) + (\boldsymbol{\chi}_i)_k(t) \tag{44}$$

   Call the set of corresponding C_α displacements $V_{\mathrm{p}} = \{\boldsymbol{v}_k^{\mathrm{p}}\}$.
6. The mode shapes in $\bar{V}$ form an orthonormal set of modes. After the transformation of step 4, and the projection of step 5, V_{p} is no longer orthonormal. To resolve this we perform the Gram–Schmidt orthogonal process. The resulting set of orthonormal mode shapes $V_{\mathrm{c}} = \{\boldsymbol{v}_i^{\mathrm{c}}\}$ is obtained by iteratively applying

$$\boldsymbol{v}_i^{\mathrm{c}} = \frac{\boldsymbol{v}_i^{\mathrm{p}} - \sum_{k=1}^{i-1} (\boldsymbol{v}_i^{\mathrm{p}} \boldsymbol{v}_k^{\mathrm{c}}) \boldsymbol{v}_k^{\mathrm{c}}}{\| \boldsymbol{v}_i^{\mathrm{p}} - \sum_{k=1}^{i-1} (\boldsymbol{v}_i^{\mathrm{p}} \boldsymbol{v}_k^{\mathrm{c}}) \boldsymbol{v}_k^{\mathrm{c}} \|} \tag{45}$$

[8] Note: The calculation of $M^{-1/2}$ is not computationally limiting because the known structure of M enables an $\mathcal{O}(n)$ implementation.

2.6.2. "Unmixing" and decomposition analysis

To simplify the following expressions let $n_{\mathrm{c}} = 6N$ (the number of cluster-NMA modes), $n_{\mathrm{s}} = 3n$ (the number of standard-NMA modes), and d be the dimension of the space used to describe all C_α displacements (note: $d = n_{\mathrm{s}}$, but separate variables are used to make other comparisons more apparent). Let the cluster-NMA mode shapes (after the operations of Section 2.6.1 have been performed) be given by

$$V_{\mathrm{c}} = [\boldsymbol{v}_1^{\mathrm{c}}, \ldots, \boldsymbol{v}_{n_{\mathrm{c}}}^{\mathrm{c}}] \in \mathbb{R}^{d \times n_{\mathrm{c}}} \tag{46}$$

and let the standard-NMA mode shapes (the harmonic solutions of Eq.(4)) be given by

$$V_{\mathrm{s}} = [\boldsymbol{v}_1^{\mathrm{s}}, \ldots, \boldsymbol{v}_{n_{\mathrm{s}}}^{s}] \in \mathbb{R}^{d \times n_{\mathrm{s}}} \tag{47}$$

Since the mode shapes of V_{s} are orthonormal (and they span $\mathbb{R}^d$) the motions of V_{c} can be uniquely decomposed over V_{s} as

$$D_{i,j} = |\boldsymbol{v}_i^{\mathrm{c}} \boldsymbol{v}_j^{\mathrm{s}}| \Rightarrow D = |[V_{\mathrm{c}}]^{\mathrm{T}}[V_{\mathrm{s}}]| \in \mathbb{R}^{n_{\mathrm{c}} \times n_{\mathrm{s}}} \tag{48}$$

The entries of D measure the alignment of the corresponding cluster and standard modes. This matrix can be broken into block form as shown in Fig. 3.

By assumption, the cluster mode shapes should be composed of, and limited to, only the low standard-NMA motions. Accordingly, the desired mode shapes are defined as

$$\tilde{\boldsymbol{v}}_i^{\mathrm{c}} = \sum_{j=1}^{m} \alpha_{i,j} \boldsymbol{v}_j^{\mathrm{c}} = [V_{\mathrm{c}}^m]\boldsymbol{\alpha}_i \tag{49}$$

for $i \in [1, \ldots, m]$ where m is the bound on the number of included cluster modes. The undetermined weighting coefficients, $\alpha_{i,j}$, are represented in vector form as $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \ldots, \alpha_{i,m}]^{\mathrm{T}}$. The first m cluster modes are the column vectors of V_{c}^m.

It is important to note that V_{c} has n_{c} linearly independent mode shapes. In practice, we choose $m < n_{\mathrm{c}}$, thus the mode shapes produced by Eq. (49) will be (by definition) contained within the span of the lowest m cluster modes. Since the lowest m cluster modes are dominantly composed of the lowest m standard modes, the resulting set of m unmixed mode shapes will more accurately reflect the desired mode shapes *and* mode ordering.

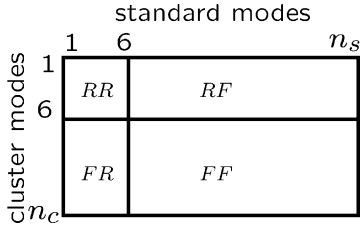

Fig. 3. Block representation of D. The sub-blocks correspond to the decompositions. RR: rigid cluster modes over rigid standard modes. RF: rigid cluster modes over flexible standard modes. FR: flexible cluster modes over rigid standard modes. FF: flexible cluster modes over flexible standard modes.

The constraint of Eq. (49) is equivalent to the change of coordinates $\boldsymbol{\sigma} = [V_c^m]\boldsymbol{\alpha}$. Application of this transform to the energy equations of the point mass model yields

$$T_c = \frac{1}{2}\dot{\boldsymbol{\sigma}}^T M_s \dot{\boldsymbol{\sigma}} = \frac{1}{2}\dot{\boldsymbol{\alpha}}^T \underbrace{[V_c^m]^T M_s [V_c^m]}_{M_s'} \dot{\boldsymbol{\alpha}} \tag{50}$$

$$V_s = \frac{1}{2}\boldsymbol{\sigma}^T K_s \boldsymbol{\sigma} = \frac{1}{2}\boldsymbol{\alpha}^T \underbrace{[V_c^m]^T K_s [V_c^m]}_{K_s'} \boldsymbol{\alpha} \tag{51}$$

which produces the m dimensional equation

$$M_s'\ddot{\boldsymbol{\alpha}} + K_s'\boldsymbol{\alpha} = \mathbf{0} \tag{52}$$

This equation represents the EOM for the structure, but rather than using a $\boldsymbol{\sigma}$ parameterization, the $\boldsymbol{\alpha}$ parameterization is used so that the undetermined coefficients of Eq. (49) can be recovered.

Since M_s and K_s are used in the unmixing process it must be stated that the creation of these $3n \times 3n$ matrices does not defeat the purpose of cluster-NMA. We are not solving for the mode shapes of this full system—we are only using M_s and K_s in matrix multiplication with much smaller dimensioned matrices.[9] Calculation of these matrices does not impose any limitations on memory or computational complexity above that which is already required by the cluster-NMA method.

Returning to Eq. (52), the change of coordinate $\boldsymbol{\beta} = [M_s']^{1/2}\boldsymbol{\alpha}$ yields

$$\ddot{\boldsymbol{\beta}} + \underbrace{[M_s']^{-1/2}[K_s'][M_s']^{-1/2}}_{\Omega = \Omega^T}\boldsymbol{\beta} = \mathbf{0} \tag{53}$$

Let the eigenvectors of Ω be V_β, which are then transformed back into $\boldsymbol{\alpha}$ coordinates as

$$V_\alpha = [M_s']^{-1/2}[V_\beta] \triangleq [\boldsymbol{v}_1^\alpha, \ldots, \boldsymbol{v}_m^\alpha] \in \mathbb{R}^{m\times m} \tag{54}$$

The undetermined coefficients as defined in Eq. (49), are thus obtained with $\boldsymbol{\alpha}_i = \boldsymbol{v}_i^\alpha$ yielding the unmixed mode shapes

$$\tilde{\boldsymbol{v}}_i^c = [V_c^m]\boldsymbol{\alpha}_i \rightarrow \tilde{V}_c^m = [V_c^m][V_\alpha] \in \mathbb{R}^{d\times m} \tag{55}$$

After this process, the set of unmixed mode shapes in $\tilde{V}_c^m$ is not orthonormal—thus, the Gram–Schmidt orthonormalization process is applied again. The final set of motions is defined as $\hat{V}_c^m = [\hat{\boldsymbol{v}}_1^c, \ldots, \hat{\boldsymbol{v}}_m^c]$ and the decomposition matrix is now given by $D = |[\hat{V}_c^m]^T[V_s]|$.

[9] A quick computational complexity verification: Let there be at most z non-zero entries in each row (or column) of K_s. The first matrix multiplication for computing K_s' is: $[V_c^m]^T K_s$, which is $\mathcal{O}(3nmz) \approx \mathcal{O}(n)$. The second matrix multiplication: [result]$[V_c^m]$ is $\mathcal{O}(3nm^2) \approx \mathcal{O}(n)$. The overall computational complexity is thus $\mathcal{O}(n)$. By the same argument with $z = 1$, we get the computational complexity of computing M_s' as $\mathcal{O}(n)$.

3. Application of cluster-NMA to various protein structures

In this section, we apply a high- and low-resolution helix-based clustering algorithm as described in Section 2.2. Cluster-NMA is tested on a sample set of 12 protein structures ranging in size from 85 to 1287 residues. An even coarser cluster-NMA is performed on lactoferrin (691 residues) by clustering by domain. Finally, a very coarse cluster-NMA is performed on the 8015 residue GroEL/GroES complex. The range in structure size and cluster resolutions is chosen so that computational savings and mode accuracy of cluster-NMA can be more fully probed.

The sample set of 12 structures range in size from 85 to 1287 residues and represent a wide range of conformations. All residue interactions are defined by a distance cut-off of 12 Å and a maximum contact number of 20. For cluster-NMA, the helix-based clustering method is applied to the helices, as identified in the PDB files. The high-resolution cluster-NMA is calculated with a target cluster size of 3 and a minimum cluster size of 3. The lower resolution analyses are performed with a target cluster size of 10 and a minimum cluster size of 5. In both cases, the lowest 26 modes are unmixed with the lowest 10 non-rigid modes expected to show strong alignment with their corresponding standard modes. The parameters and computation times are given in Table 1.

We now consider two specific structures whose mode decompositions are representative of the result types produced by high-resolution cluster-NMA. Each data table shows the upper left 10×10 block of *FF*. The most aligned standard mode with each cluster mode is given in bold (i.e. one bold value per row). The vector norm of each column is also given below the corresponding column. The vector norms indicate how well each standard mode (corresponding column) is captured by the lowest 10 non-rigid cluster modes.

Table 2 shows a decomposition with excellent alignment values, perfect mode ordering over the lowest 10 non-rigid modes, and almost perfect column vector norms. This almost ideal decomposition is produced by 8 of the 12 example structures.

The worst decomposition of all sample structures is shown in Table 3. The alignment values are still excellent over the lowest five non-rigid modes, but modes 12 and 13 have merged in almost equal parts and modes 14 and 15 have swapped in order. Since cluster-NMA identifies mode shapes by frequency on a constrained version of the full structure, standard modes with similar frequencies may occasionally disrupt the ordering of the cluster modes.

If the global motions are of interest, then the mode swapping and mode merging are not of importance because, as indicated by the column vector norms, each standard mode is still very well captured by only the set of the lowest 10 non-rigid cluster modes. If the desired application requires exact mode shapes then the cluster-NMA mode shapes

Table 1
Set of sample structures comparing the lower- and higher-resolution cluster-NMA results

PDB	Size	C_α-NMA	Cluster-NMA ($t = 3$, $m = 3$)			Cluster-NMA ($t = 10$, $m = 5$)		
		Time	Clusters	Time	Ratio	Clusters	Time	Ratio
1A32	85	1.3	24	6.7	0.19	8	6.6	0.24
1CLL	144	3.3	45	11.2	0.29	14	11.7	0.40
1AKY	218	9.2	66	19.0	0.49	19	19.6	0.62
1A54	321	25.7	98	35.1	0.73	29	31.4	1.01
3ICD	414	50.6	129	54.9	0.92	35	45.8	1.28
1DDT	523	94.6	159	81.4	1.16	50	64.2	1.72
1LFH	691	209.2	214	150.1	1.39	64	91.4	2.62
1E18	779	296.5	234	188.9	1.57	76	149.1	2.42
1H3N	814	323.9	248	212.4	1.53	76	155.1	2.52
1RUX	884	420.0	280	275.2	1.53	81	144.8	3.14
1EUL	994	568.9	304	340.0	1.67	93	166.0	3.79
1EPW	1287	1223	402	683.1	1.79	124	233.5	5.45

All computations are done in MATLAB 12.1 on a 1.6 GHz Intel Pentium 4M with 512 MB of RAM.

Table 2
Decomposition matrix for PDB: 1EUL

Cluster	Standard									
	7	8	9	10	11	12	13	14	15	16
7	**0.9996**	0.0246	0.0040	0.0052	0.0025	0.0049	0.0040	0.0020	0.0000	0.0005
8	0.0244	**0.9991**	0.0222	0.0096	0.0141	0.0105	0.0084	0.0064	0.0026	0.0056
9	0.0043	0.0218	**0.9988**	0.0363	0.0033	0.0085	0.0075	0.0068	0.0058	0.0044
10	0.0053	0.0097	0.0352	**0.9971**	0.0018	0.0567	0.0157	0.0049	0.0068	0.0096
11	0.0035	0.0152	0.0029	0.0071	**0.9938**	0.0969	0.0313	0.0087	0.0088	0.0008
12	0.0055	0.0101	0.0099	0.0537	0.0994	**0.9886**	0.0872	0.0124	0.0121	0.0099
13	0.0034	0.0065	0.0094	0.0206	0.0228	0.0880	**0.9936**	0.0141	0.0236	0.0236
14	0.0018	0.0053	0.0058	0.0059	0.0104	0.0098	0.0124	**0.9894**	0.1180	0.0610
15	0.0001	0.0039	0.0056	0.0036	0.0061	0.0164	0.0187	0.1229	**0.9791**	0.1296
15	0.0005	0.0052	0.0044	0.0100	0.0047	0.0053	0.0225	0.0433	0.1289	**0.9807**
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0.9999	0.9999	0.9998	0.9995	0.9992	0.9991	0.9986	0.9982	0.9950	0.9915

This decomposition has very strong alignment values and exact mode ordering.

can be refined by using an iterative power method with K_s. Such a method typically starts with an initial vector (the candidate cluster-NMA mode shape) and iterates until the resulting sequence of vectors converges. With the newly refined, normalized mode shape, $\hat{\boldsymbol{v}}$, the corresponding eigenvalue is easily obtained with $\hat{\lambda} = \|[K_s]\hat{\boldsymbol{v}}\|$. The set of refined mode shapes can then be reordered according to their refined eigenvalues.

Table 3
Decomposition matrix for PDB: 1AKY

Cluster	Standard									
	7	8	9	10	11	12	13	14	15	16
7	**0.9960**	0.0029	0.0266	0.0019	0.0363	0.0175	0.0087	0.0032	0.0030	0.0028
8	0.0032	**0.9899**	0.0477	0.0701	0.0058	0.0565	0.0242	0.0308	0.0062	0.0108
9	0.0280	0.0396	**0.9878**	0.0418	0.0150	0.0776	0.0160	0.0085	0.0271	0.0128
10	0.0041	0.0512	0.0229	**0.9391**	0.1995	0.2226	0.0736	0.0313	0.0142	0.0304
11	0.0320	0.0196	0.0043	0.2146	**0.9530**	0.0940	0.1045	0.0265	0.0282	0.0037
12	0.0188	0.0176	0.0587	0.0547	0.1523	*0.6159*	**0.7341**	0.1365	0.0522	0.0221
13	0.0021	0.0597	0.0438	0.1862	0.0519	**0.6736**	*0.6307*	0.1274	0.0674	0.1072
14	0.0030	0.0051	0.0303	0.0190	0.0330	0.0557	0.0349	0.2519	**0.8992**	0.2252
15	0.0002	0.0202	0.0166	0.0182	0.0070	0.1114	0.0458	**0.7129**	0.3036	0.5128
16	0.0047	0.0399	0.0197	0.0490	0.0248	0.1591	0.0754	0.4238	0.0564	**0.6275**
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0.9971	0.9952	0.9931	0.9876	0.9885	0.9705	0.9814	0.8881	0.9555	0.8489

Modes 12 and 13 have merged and modes 14 and 15 have swapped order. The italic values highlight the parts of modes 12 and 13 that are correctly ordered, even though the larger (bold value) in each of these rows is off the diagonal.

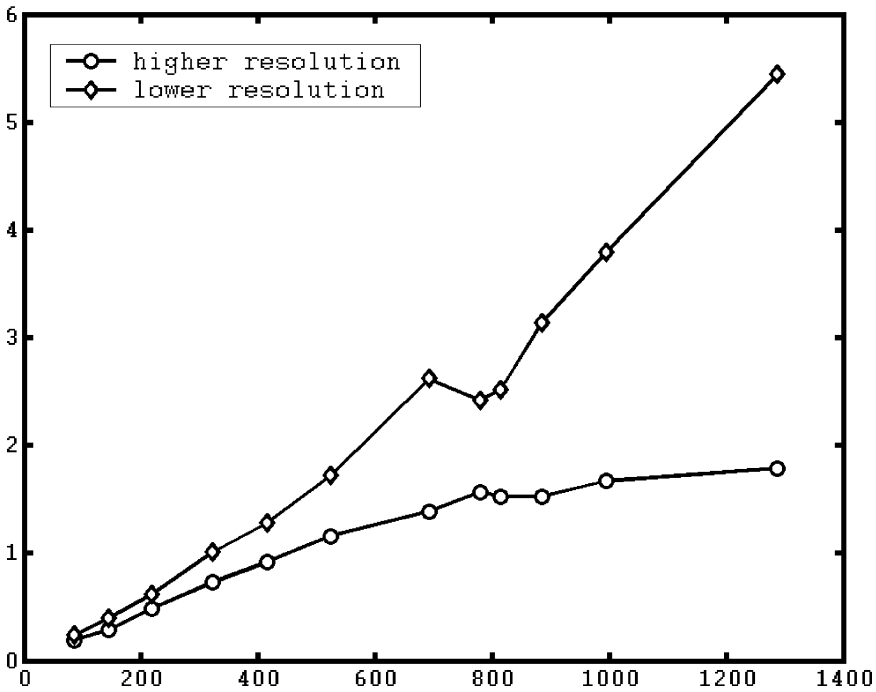

Fig. 4. Plot showing the ratio between C_α-NMA and cluster-NMA computation times as a function of structure size for lower- and higher-resolution cluster-NMA.

We now reconsider lactoferrin with a domain-based clustering. The five clusters are defined as {1–91, 250–320}, {92–249}, {321–332}, {333–436, 594–691}, and {437–593}. This 30 DOF model uses less than 1.5% as many DOFs as the all C_α model. The mode decomposition matrix, as expected, is not as tight as those seen in higher-resolution cluster-NMA. However, this very coarse model still captures the global motion very well. The corresponding column vector norms over the lowest 10 modes are 0.98, 0.99, 0.98, 0.97, 0.96, 0.94, 0.93, 0.90, 0.86, 0.61.

The final example structure we consider is the GroEL/GroES complex (PDB: 1AON). This structure is composed of 14 repeating chains of 524 residues and 7 repeating chains of 97 residues. Clustering by chain requires 126 DOFs which is approximately 0.5% as many DOFs as required by the all C_α model. By using the all C_α model computation times from Table 1, we can extrapolate out to determine that all C_α NMA on the GroEL/GroES complex would take approximately 77 h (neglecting the substantial memory limitations). The cluster-NMA took less than 3 h, for a savings factor of approximately 26.5. The mode shapes satisfactorily agree with other qualitative descriptions in the literature [24].

We now address computational performance. Fig. 4 shows the ratio of NMA computation times (C_α/cluster) as a function of structure size for the higher- and lower-resolution clusterings from Table 1. The linear plot for the lower resolution case indicates that cluster-NMA performs an entire order of magnitude better than all C_α NMA.

4. Conclusions

At the core of cluster-NMA is the rigid-body representation of the protein structure. This simultaneously reduces the number of DOFs and confines the structure to the space of low-frequency motions. Typically, NMA computational performance is limited by the $\mathcal{O}(n^3)$ eigenproblem. Cluster-NMA circumvents this limitation by using an $\mathcal{O}(n)$ transformation to project the structure into a reduced DOF representation. The eigenproblem is then performed in this smaller space and the results are transformed back to the full DOF representation with a final $\mathcal{O}(n)$ transformation.

To make exact comparisons of various mode shapes at residue resolution the transformation back to full DOF representation is necessary. However, if only mode visualization is of interest the second transformation is not necessary. The protein structure can be represented by its set of rigid clusters which do not need to be shown in full residue detail (their outer surfaces are sufficient). The mode shapes of the reduced DOF representation directly specify the translational and rotational motion of each cluster and thus serve as an efficient normal mode or global mode visualization tool.

In this paper, cluster-NMA is applied uniformly at varying levels of resolution over the entirety of each of the 12 sample structures. In application, it may only be desirable to study the dynamics of smaller regions within a large structure. In such cases, clustering can be made fine in the

regions of interest and very coarse elsewhere (i.e. spend the DOFs only on the regions of interest). This flexibility of application greatly enhances the computational performance of cluster-NMA, as seen in the coarse clustering of lactoferrin and the GroEL/GroES complex.

Cluster-NMA can be effectively used to capture motions consistent with the low mode shapes of an all C_α model. Cluster-NMA also very accurately captures the span of the low-frequency standard modes as indicated by the high column vector norms (even in the case of mode swapping and/or merging). These results make cluster-NMA an ideal tool for efficiently calculating global harmonic motions of very large structures about an equilibrium conformation.

References

[1] N. Gō, A theorem on amplitudes of thermal atomic fluctuations in large molecules assuming specific conformations calculated by normal mode analysis, Biophys. Chem. 35 (1990) 105–112.

[2] T. Ha, Single-molecule fluorescence resonance energy transfer, Methods 25 (2001) 78–86.

[3] A.G. Palmer III, Probing molecular motion by NMR, Curr. Opin. Struct. Biol. 7 (1997) 732–737.

[4] S. Doniach, P. Eastman, Protein dynamics simulations from nanoseconds to microseconds, Curr. Opin. Struct. Biol. 9 (1999) 157–163.

[5] K. Hinsen, Analysis of domain motions by approximate normal mode calculations, Proteins: Struct. Function Genet. 33 (1998) 417–429.

[6] K. Hinsen, Domain motions in proteins, J. Mol. Liquids 84 (2000) 53–63.

[7] I. Bahar, A.R. Atilgan, M.C. Demirel, B. Erman, Vibrational dynamics of folded proteins: significance of slow and fast motions in relation to function and stability, Phys. Rev. Lett. 80 (12) (1998) 2733–2736.

[8] A. Amadei, A.B.M. Linssen, H.J.C. Berendsen, Essential dynamics of proteins, Proteins: Struct. Function Genet. 17 (4) (1993) 412–425.

[9] O. Keskin, S.R. Durell, I. Bahar, R.L. Jernigan, D.G. Covell, Relating molecular flexibility to function: a case study of tubulin, Biophys. J. 83 (2002) 663–680.

[10] I.Bahar, A.R. Atilgan, B. Erman, Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential, Fold. Design 2 (3) (1997) 173–181.

[11] I. Bahar, A. Wallqvist, D.G. Covell, R.L. Jernigan, Correlation between native-state hydrogen exchange and cooperative residue fluctuations from a simple model, Biochemistry 37 (1998) 1067–1075.

[12] T. Haliloglu, I. Bahar, Structure-based analysis of protein dynamics: comparison of theoretical results for hen lysozyme with X-ray diffraction and NMR relaxation data, Proteins: Struct. Function Genet. 37 (1999) 654–667.

[13] A.R. Atilgan, S.R. Durell, R.L. Jernigan, M.C. Demirel, O. Keskin, I. Bahar, Anisotropy of fluctuation dynamics of proteins with an elastic network model, Biophys. J. 80 (2001) 505–515.

[14] M. Kim, G.S. Chirikjian, R.L. Jernigan, Elastic models of conformational transitions in macromolecules, J. Mol. Graphics Modell. 21 (2002) 151–160.

[15] M.K. Kim, R.L. Jernigan, G.S. Chirikjian, Efficient generation of feasible pathways for protein conformational transitions, Biophys. J. 83 (2002) 1620–1630.

[16] P. Doruker, R.L. Jernigan, I. Navizet, R. Hernandez, Important fluctuation dynamics of large protein structures are preserved upon coarse-grained renormalization, Int. J. Quant. Chem. 90 (2002) 822–837.

[17] F. Tama, F.X. Gadea, O. Marques, Y.-H. Sanejouand, Building-block approach for determining low-frequency normal modes of macromolecules, Proteins: Struct. Function Genet. 41 (2000) 1–7.

[18] H.M. Chun, C.E. Padilla, D.N. Chin, M. Watanabe, V.I. Karlov, H.E. Alper, K. Soosaar, K.B. Blair, O.M. Becker, L.S.D. Caves, R. Nagle, D.N. Haney, B.L. Farmer, MBO(N)D: a multibody method for long-time molecular dynamics simulations, J. Comput. Chem. 21 (3) (2000) 159–184.

[19] M. Tirion, Large amplitude elastic motions in proteins from a single-parameter, atomic analysis, Phys. Rev. Lett. 77 (9) (1996) 1905–1908.

[20] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progression, J. Symbolic Comput. 9 (1990) 251–280.

[21] D.J. Jacobs, A.J. Rader, L.A. Kuhn, M.F. Thorpe, Protein flexibility predictions using graph theory, Proteins: Struct. Function Genet. 44 (2001) 150–165.

[22] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank.

[23] G.S. Chirikjian, A.B. Kyatkin, Engineering Applications of Noncummutative Harmonic Analysis, CRC Press, Boca Raton, 2001.

[24] O. Keskin, I. Bahar, D. Flatow, D.G. Covell, R.L. Jernigan, Molecular mechanisms of chaperonin GroEL-GroES function, Biochemistry 41 (2002) 491–501.