

# Parameterizing Elastic Network Models to Capture the Dynamics of Proteins

Patrice Koehl, Henri Orland, Marc Delarue

▶ **To cite this version:**

Patrice Koehl, Henri Orland, Marc Delarue. Parameterizing Elastic Network Models to Capture the Dynamics of Proteins. Journal of Computational Chemistry, 2021, 42 (23), pp.1643-1661. 10.1002/jcc.26701 . pasteur-03413465

**HAL Id: pasteur-03413465**

**https://pasteur.hal.science/pasteur-03413465**

Submitted on 3 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Parameterizing Elastic Network Models to Capture the Dynamics of Proteins

Patrice Koehl *, Henri Orland †, Marc Delarue ‡

November 3, 2021

**Abstract**

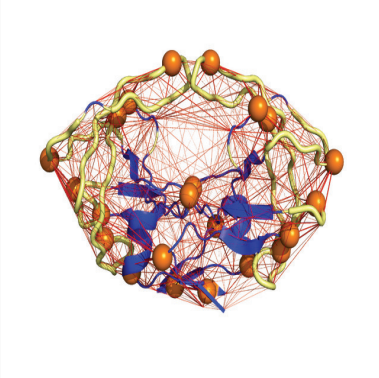
Coarse-grained normal mode analyses of protein dynamics rely on the idea that the geometry of a protein structure contains enough information for computing its fluctuations around its equilibrium conformation. This geometry is captured in the form of an elastic network (EN), namely a network of edges between its residues. The normal modes of a protein are then identified with the normal modes of its EN. Different approaches have been proposed to construct ENs, focusing on the choice of the edges that they are comprised of, and on their parameterizations by the force constants associated with those edges. Here we propose new tools to guide choices on these two facets of EN. We study first different geometric models for ENs. We compare cutoff-based ENs, whose edges have lengths that are smaller than a cutoff distance, with Delaunay-based ENs and find that the latter provide better representations of the geometry of protein structures. We then derive an analytical method for the parameterization of the EN such that its dynamics leads to atomic fluctuations that agree with experimental B-factors. To limit overfitting, we attach a parameter referred to as flexibility constant to each atom instead of to each edge in the EN. The parameterization is expressed as a non-linear optimization problem whose parameters describe both rigid-body and internal motions. We show that this parameterization leads to improved ENs, whose dynamics mimic MD simulations better than ENs with uniform force constants, and reduces the number of normal modes needed to reproduce functional conformational changes.

Keywords: Elastic network models, coarse-grained normal modes, protein dynamics, b-factors, rigidity

*Department of Computer Sciences and Genome Center, University of California, Davis, CA 95616, USA

†Institut de Physique Théorique, Université Paris-Saclay, CEA, 91191 Gif sur Yvette, France

‡Unité de Dynamique Structurale des Macromolécules, UMR 3528 du CNRS, Institut Pasteur, 75015 Paris, France



The elastic network model of a protein is a network of springs (red) whose dynamics is expected to mimic the dynamics of the protein. This is achieved if the network is properly parameterized. We develop a mathematical approach in which experimental atomic fluctuations serve to generate this parameterization. Using the parameterized elastic network, we identify the rigid (blue) and flexible (yellow) regions in a protein, as well as the essential residues for its dynamics (orange).

# 1 INTRODUCTION

The function of a biomolecule derives from the specific dynamics of its structure. The need to observe and analyze such dynamics is therefore at the core of many studies in structural molecular biology. Unfortunately, we currently lack the experimental and computational tools for a comprehensive representation of the dynamics from the molecular to supra-molecular levels. Indeed, only a few experimental techniques can collect time-resolved structural data, and those that can are usually limited to a narrow time range. In parallel, current computational methods such as atomistic molecular dynamics simulations are restricted in scope, both for time-scale (usually micro-seconds) and length-scale (with systems of up to hundred thousand atoms), because of limitations in computing power. To circumvent such problems, there is a need to develop simplified, albeit accurate models to study the dynamics of a molecule on a computer, to inform those models based on available experimental data, and to assess their relevance, correctness and usefulness. In this paper, we address some of these issues in the context of coarse-grained normal mode analyses of biomolecular dynamics based on elastic network models (ENM).

***Experimental data on protein dynamics.*** Proteins are not static objects and occupy instead ensemble of conformations. Dynamics is the study of the kinetics of the transitions between these states. They may occur on a global scale, as observed in allostery or catalysis, or at a local scale, the so-called local flexibility. Evidence of such local flexibility is obtained either from NMR spectroscopy, by analyzing the spin relaxation of individual atoms and assigning them an order parameter, or from X-ray crystallography, by assigning and refining a B-factor, also called the Debye-Waller factor, to each atom to account for their mobility in the crystal (see for example[1]). Both have proved useful for analyzing protein dynamics (see for example Ref.[2] for the use of B-factors, and Ref.[3] for the use of order parameters). As such, there have been many efforts to predict their values from the knowledge of the static structure of a protein[4–9], from the sequence of the protein[10,11], or from of the dynamics of the proteins, either derived from rigid motions[12], from molecular dynamics simulations[13–15], or from normal mode analyses[16–20]. It is worth noting that both the NMR order parameters and the crystallographic B-factors are not quantities that are

directly observed from an experiment. The NMR order parameter $\mathcal{S}^2$ is derived from the so-called model-free approach introduced by Lipari and Szabo[21,22], in which the motion of an atom is described as the combination of overall rotational reorientation characterized with a correlation time $\tau_c$ and internal motions described with an amplitude, the order parameter. In parallel, a B-factor is a parameter that is introduced to account for atomic displacements during the data collection as well as conformational differences in the different unit cells of the crystal after plunging (freezing) it in liquid nitrogen. As such, it is dependent on the conditions in which those data were collected, if the crystals were frozen in liquid nitrogen or not, as well as on the refinement process of these data to derive a structure. As such B-factors of one crystal structure cannot be directly compared to those of another. Despite those limitations, as mentioned above, both NMR order parameters and X-ray B-factors remain important source of information on the dynamics of proteins. In this paper, we focus on B-factors.

***Computational approaches to studying protein dynamics.*** Probably the most natural approach to studying protein dynamics on a computer is to assume that this dynamics follows classical mechanics and accordingly to solve Newton's equations at the atomic level: this is the idea behind the now ubiquitous molecular dynamics simulations. However, such simulations are computationally demanding, and despite progress in hardware, software, and representations of the molecular system, there is an interest in developing alternate approaches that would be applicable even on commodity computers. A promising approach is to infer dynamics from static structures corresponding to locally stable states[23], together with reliable coarse-graining approaches to bridge the time-scale gap[24,25]. Normal Modes, for example, represent a class of movements around a local energy minimum that have been found in many instances to be biologically relevant[26–30]. Normal modes based on traditional force fields can, however, be relatively difficult to compute, as those forcefields include terms such as the vdW interactions that are not well approximated with a quadratic term. The Elastic Network Model (ENM), introduced by M. Tirion in 1996, offers a particularly simple and efficient way to circumvent this problem by building a geometric, quadratic potential with the experimental structure as its minimum, allowing fast access to the collective dynamics of even large protein complexes[31]. Tirion validated her model by showing that its

low frequency modes match well with those computed from traditional normal modes on G-actin. Her observation has been confirmed multiple times since then. Coarse grained normal mode analyses (NMA) based on the ENM have proved useful to characterize the allosteric change in conformation undergone by hemoglobin from its tense (T) form to its relaxed (R) form[32], to analyse conformational transitions in DNA-based poymerases[33], to analyze global ribosome motions[34], and to study the dynamics of viral capsids[35–38], among others. Such coarse-grained analyses of biomolecular dynamics have developed as a viable alternative to traditional molecular dynamics simulations[23,39–42]. It should be noted that NMA have proved also useful in structure refinements based on experimental studies in which dynamics is considered, such as X-ray crystallography[43,44] and cryo-electron microscopy[45–47].

***The physical model behind EN models.*** Two categories of normal mode analyses based on ENMs are widely used today, namely, the Gaussian Network Model (GNM)[48,49] and the anisotropic network model (ANM)[17,31,50]. Here we follow the latter model, in which the energy of a molecule is equated with the harmonic energy associated with springs attached to a set of pairs of atoms. This defines a quadratic energy on the inter-atomic distances,

$$V(\mathbf{X}) = \frac{1}{2}\sum_{(i,j)} k_{ij}(r_{ij} - r_{ij}^0)^2 \tag{1}$$

when the biomolecule is in conformation $\mathbf{X}$. In this equation, $k_{ij}$ is the force constant of the "spring" formed by the pair of atoms $i$ and $j$, and $r_{ij}$ and $r_{ij}^0$ are the distances between $i$ and $j$ in the conformation $\mathbf{X}$, and in the reference conformation $\mathbf{X}^0$, usually taken to be the crystal structure. This model is quite simple as it relies on a very small number of coarse-grained parameters. As such, it allows for easy computations of coarse-grained normal modes (this will be discussed in the next section). There are, however, two important decisions to make when choosing those parameters that shape the model and consequently influence its effectiveness. First, the geometry of the EN needs to be specified. The potential $V$ involves a sum over pairs of atoms $(i, j)$. These pairs can be selected as those that satisfy a cutoff criterium, or as the pairs that best describe the geometric structure of the molecule. Second, values need to be assigned to the force constants $k_{ij}$ associated with those pairs of atoms. In this paper, we study both decisions. They are discussed in the two following paragraphs.

***The geometry of EN models.*** Several criteria can be used to define the set of atom

pairs that are used in Eq. (1). In standard EN, the criterium is usually a cutoff distance $R_c$ such that atoms separated by less than this cutoff are included in the EN. There are however no guidelines as to which values for $R_c$ are best and sometimes different implementations lead to contradicting optimal values. Typical values for $R_c$ within ANM models are in the range 13-15 Å when the ENM is based on $C_\alpha$ only[51]. To avoid selecting a cutoff, it has also been proposed to include all pairs of residues and to assign length-dependent force constants to their corresponding springs. For example, Hinsen[52] and Kovacs et al[53] used force constants with exponential distance-dependence, while Yang et al.[54] developed a parameter-free ENM in which the force constants are inversely scaled by the squared distance of separation. Note that in all those approaches, even those based on cutoff, a larger number of interactions are considered. An alternate method is to build a geometric structure on the sets of positions of the atoms; the Delaunay complex and its subsequent alpha shape filtrations are well suited for this purpose[55].

***Parameterizing the force constants.*** The choice of the values for the force constants is also important, as they define the amplitude of the predicted internal motions of the molecule of interest. In her original EN model, Tirion set the force constants to be equal for all pairs of atoms in the ENM and selected this value such that the density of ANM modes matches with the density of normal modes computed on the same molecule with a traditional force field[31]. Nowadays, the trend is to derive the scale of the force constants by fitting the predicted thermal displacements of each atom to the experimental mean square fluctuations, namely the B-factors in X-ray crystallography. Assuming different force constants for each interactions in the ENM, and assuming that internal motions dominate the dynamics detected with B-factors, perfect fits can be obtained[55,56]. There is a danger of over-fitting[57], however, as the number of force constants is significantly larger than the number of experimental values used for the fit. In addition, the implicit assumption of the dominance of internal motions has been questioned. It is known that B-factors are also influenced by rigid-body motions taking place in the crystal[12]. In addition, molecules in crystal experience a different environment than when isolated in solution, and inter-atomic contacts established in the crystal have also been shown to affect the normal modes[58,59], although most likely to a lesser extent than rigid body motions[60–62].

***Our contribution.*** Our goal in this paper is to derive a method that combines the experimental information on the geometry of a protein structure (i.e. its crystallographic structure) with the dynamics information encoded in the B-factors associated with that structure to build a better elastic network model for that protein and therefore to derive a better model of its dynamics. This approach deviates from standard coarse grained normal mode models based on EN. Indeed, in our approach we build a specific model for each protein structure of interest, while standard models are designed with generic parameters that can be transferred from one protein to another. In addition, while those standard models are often used to predict B-factors, our method takes those B-factors as input. While we lose transferability, we will show that our dynamic-based EN leads to normal modes that better match with molecular dynamics simulations than normal modes derived from generic EN models.

Our approach accounts for both elements that define an EN, namely its geometry and the parameterization of its edges, as discussed above. Instead of defining the ENM using a cutoff for distance pairs, we construct the Delaunay complex over the positions of the $C_\alpha$ of the protein of interest. This construction is completely parameter free. We then assign to each $C_\alpha$ a flexibility constant $k_i$, and compute the force constant of a pair $(i, j)$ in the ENM as the harmonic mean $k_{ij} = \sqrt{k_i k_j}$ of their flexibilities. The flexibility constants are obtained from a fit to the B-factors that accounts for rigid and internal motions. The implementation and validation of this approach is a result of the four following goals that are discussed in detail in the paper:

- Establish mathematically the fitting procedure,
- Evaluate the normal modes computed from the fitted force constants by quantifying their agreement with molecular dynamics simulations,
- Analyze the amino acid specificity of the flexibility constants, and
- Characterize the concept of flexibilities in the context of the rigidity theory of proteins[63,64].

The paper is organized as follows. In the next section we provide background on NMA

and describes our fitting procedure for computing atomic flexibility based on experimental B-factors. In the methods section we describe the datasets and methods of analyses used in our numerical experiments that are described in the following section. We conclude with a general discussion on how to best parameterize coarse-grained models to compute biologically relevant normal modes.

# 2 METHODOLOGY

## 2.1 Coarse grained Normal mode analysis based on the Tirion elastic network model

Let $B$ be a protein containing $N$ atoms, with atom $i$ characterized by its position $X_i = (X_{i1}, X_{i2}, X_{i3})$. The whole molecule is then described by a $3N$ position vector $\mathbf{X}$. For two atoms $i$ and $j$ of $B$, we set $r_{ij} = |X_i - X_j|$ and $r^0_{ij} = |X^0_i - X^0_j|$ to be the Euclidean distances between them in a conformation $\mathbf{X}$ and in the ground-state conformation $\mathbf{X}^0$ (which will be taken to be the X-ray structure), respectively. The elastic potential $V$ of the biomolecule is given by equation 1. In the normal mode framework, this potential is approximated with a second-order Taylor expansion in the neighborhood of the ground state $X^0$:

$$V(\mathbf{X}) \approx \quad V(\mathbf{X}^0) + \nabla V(\mathbf{X}^0)^T(\mathbf{X} - \mathbf{X}^0) + \frac{1}{2}(\mathbf{X} - \mathbf{X}^0)^T H(\mathbf{X} - \mathbf{X}^0) \tag{2}$$

where $\nabla V$ and $H$ are the gradient and Hessian of $V$, respectively. Note that based on Equation 1, $V(\mathbf{X}^0) = 0$ and $\nabla V(\mathbf{X}^0) = 0$. The approximate elastic potential is then simply

$$V(\mathbf{X}) \approx \frac{1}{2}(\mathbf{X} - \mathbf{X}^0)^T H(\mathbf{X} - \mathbf{X}^0) \tag{3}$$

For simplicity, we will assume in the following that each atom is assigned a mass of 1. The procedure can easily be expanded to account for the exact masses of the different atom types. In Cartesian coordinates, the equations of motion defined by the potential $V$ are derived from Newton's equation:

$$\frac{d^2\mathbf{X}}{dt^2} = -H(\mathbf{X} - \mathbf{X}^0) \tag{4}$$

Writing the solution to this equation as a linear sum of intrinsic motions (the "normal modes" of the system), the trajectory of atom $i$ can be written as

$$X_i(t) = \sum_{k=1}^{3N} A_{ik}\alpha_k cos(\omega_k t + \delta_k) \tag{5}$$

we get a standard eigenvalue problem,

$$HE = E\Omega \tag{6}$$

The eigenfrequencies $\omega$ are given by the elements of the diagonal matrix $\Omega$, namely $\omega_k^2 = \Omega(k,k)$. The eigenvectors are the columns of the matrix $E$, and the amplitudes and phases, $\alpha_k$ and $\delta_k$, are determined by initial conditions. Because of the invariance of the potential $V$ to rotations and translations, the first six eigenvalues of the matrix $H$ are equal to 0.

## 2.2 Generating the elastic network

The main idea behind the concept of ENM is to define a network of harmonic springs that capture the geometry and dynamics of the molecule of interest. In the original ENM defined by Tirion[31], the network is defined as a set of links, with a link between two residues only if the distance between their $C_\alpha$ atoms is smaller than a given cutoff value $R_c$. There are however no guidelines as to which value for $R_c$ is best. Recently one of us proposed an alternate approach for filtering the set of all possible pairs using the concepts of alpha shapes and Delaunay triangulation[65]. More specifically, it was found that the set of edges included in the Delaunay triangulation of the atoms of a molecule forms an elastic network model that leads to good fit between the dynamics described by its normal modes and the experimental B-factors[55]. We briefly describe the procedure for generating the Delaunay triangulation; more details can be found in[65–67].

**Delaunay construction** Let us define a set $P$ of N points such that $P_i$ is positioned at the location of the $C_\alpha$ atom of residue $i$ in the protein $B$. We define the square distance $\pi_i(x)$ between a point $x$ and a point $P_i$ to be simply the square of the Euclidean distance, $\pi_i(x) = ||x - P_i||^2$. The Voronoi region $V_i$ of the point $P_i$ consists of all points $x$ that are at least as close to $P_i$ as to any other point in $P$, i.e. $V_i = \{x \in \mathbb{R}^3 | \pi_i(x) \leq \pi_j(x) \forall j \neq i\}$. $V_i$ is a

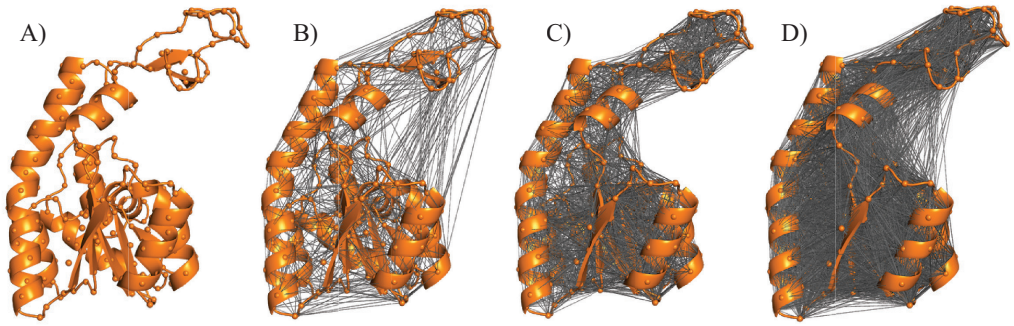



Figure 1: Illustration of a uniform force constant ENM of the adenylate kinase (PDB code 4AKE). A) Cartoon representation of the protein, with the $C_\alpha$ atoms shown as spheres. Elastic networks (gray bonds) based on a Delaunay construct (B), or a cutoff of 14 Å(C), and 20 Å(20). Those networks contain 1478, 3868, and 7863 edges, respectively.

convex polyhedron obtained as the common intersection of finitely many closed half-spaces, one per point $P_j \neq P_i$. The union of all Voronoi regions defines the Voronoi diagram of the set of points; this union covers the whole space. The Delaunay triangulation DT is the dual of the Voronoi diagram. It contains all points in $P$. In addition, we draw an edge between two points $P_i$and $P_j$ if the two corresponding Voronoi regions share a common face, called a Voronoi plane. Such an edge is included in the Delaunay triangulation. Furthermore, we draw a triangle connecting $P_i$, $P_j$, and $P_k$ if their respective $V_i$, $V_j$, and $V_k$ intersect in a common line segment, called a Voronoi edge; similarly we draw a tetrahedron between four points if their Voronoi regions meet at a common point, called a Voronoi point. Assuming general position of the points, there are no other cases to be considered: this is a central property of the Delaunay triangulation. Note that for the ENM, we only consider the edges of the Delaunay triangulation.

In the following, we will represent an ENM as $N = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges in the network, respectively. Examples of such networks generated either with a cutoff, or with the Delaunay construct, are shown in Figure 1.

## 2.3 Parameterizing the elastic network

Each edge $v_{ij} \in \mathcal{V}$ is assigned a force constant $k_{ij}$. The number of such force constants, i.e. $|\mathcal{V}|$ can be large, and usually significantly larger than the number of vertices $|\mathcal{V}|$ in the network. As our intent is to parameterize those force constants using experimental values on the vertices, we believe that allowing the former to be independent variables would lead to severe risks of overfitting. Instead, we assign to each atom (vertex) $i$ a flexibility constant, $k_i$, and define the force constant of an edge $v_{ij}$ as

$$k_{ij} = \sqrt{k_i k_j} \tag{7}$$

i.e. the geometry mean of the individual flexibility constants. An advantage of using the geometric mean is that by construction, the force constants $k_{ij}$ are positive.

## 2.4 The Hessian and its derivatives

We introduced recently a simplified representation of the Hessian of the quadratic potential defined in equation 1[38,68]. We present it here briefly, as it is relevant to our procedure for fitting B-factors.

Let us rewrite the quadratic potential for the elastic network as:

$$V(X) = \frac{1}{2} \sum_{(i,j)} V_{ij}(X) \tag{8}$$

where the summation extends to all pairs of atoms $(i, j)$ that satisfy the cutoff criterium (see above). We compute the derivatives and Hessian of this potential in vector form.

We first introduce some notations. We write the inner and outer products of two vectors $\mathbf{u}$ and $\mathbf{v}$ as $(\mathbf{u}, \mathbf{v})$ and $\mathbf{u} \otimes \mathbf{v}$, respectively. We define the vector $\mathbf{U}_{ij}$ such that $\mathbf{U}_{ij} = (0, \ldots, 0, \frac{\mathbf{x_i} - \mathbf{x_j}}{r_{ij}}, 0, \ldots, 0, \frac{\mathbf{x_j} - \mathbf{x_i}}{r_{ij}}, 0, \ldots, 0)$, namely $\mathbf{U}_{ij}$ is zero everywhere, except at positions $i$ and $j$ where it is equal to the normalized difference vector between the positions of $i$ and $j$.

Let us first analyze the pairwise potential $V_{ij}(X)$. Its gradient in $\mathbb{R}^{3N}$ at a position $\mathbf{X}$ is given by:

$$\nabla V_{ij}(\mathbf{X}) = k_{ij}(r_{ij} - r_{ij}^0)\mathbf{U}_{ij} \tag{9}$$

and its Hessian at the same position $\mathbf{X}$ is given by:

$$H_{ij}(\mathbf{X}) = k_{ij}(r_{ij} - r_{ij}^0)\frac{\delta \mathbf{U}_{ij}}{\delta \mathbf{X}} + k_{ij}\mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{10}$$

Note that both terms in the expression of the Hessian are matrices of size $3N \times 3N$. For normal mode analyzes, the gradient and Hessian are evaluated at $\mathbf{X}^0$:

$$\nabla V_{ij}(\mathbf{X}^0) = \mathbf{0} \tag{11}$$

and

$$H_{ij}(\mathbf{X}^0) = k_{ij}\mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{12}$$

The total Hessian of the elastic potential is then given by:

$$H = H(\mathbf{X}^0) = \sum_{(i,j)} k_{ij}\mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{13}$$

In this equation, the vectors $\mathbf{U}$ only depend on the ground state conformation of the molecule, and not on the force constants $k$. The derivatives of the Hessian with respect to any of those $k_{ij}$ are then trivially given by

$$\frac{dH}{dk_{ij}} = \mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{14}$$

Using the chain rule, the derivatives of the Hessian with respect to the flexibility constants $k_i$ are then,

$$\frac{dH}{dk_i} = \sum_{j|(ij)\in\mathcal{V}} \frac{k_j}{2k_{ij}}\mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{15}$$

where the summation extends over all edges that include $i$. Note that we have assumed that $k_{ij}$ is non-zero, i.e. that an edge is included in the ENM if and only if it actually contributes to the dynamics.

Expressing the Hessian as a (weighted) sum of tensor products (Equation 13) has the additional advantages of reducing the amount of memory required to store the Hessian, and to provide for simpler computations of Hessian-vector multiplications[68].

## 2.5 Calibration of the force constants using the experimental B-factors

### 2.5.1 Experimental fluctuations

In X-ray crystallography, the B-factor, or Debye-Waller factor, describes the attenuation of x-ray scattering caused by thermal motion. The isotropic B-factor of an atom $i$ is related to its positional fluctuation $< |\Delta X_i|^2 >$ by

$$B_i^{exp} = \frac{8\pi^2}{3} < |\Delta X_i|^2 > \tag{16}$$

where the brackets indicate time averages.

#### Complete fit

Equation 16 gives us a way to relate the experimental B-factor to fluctuations observed in dynamics simulation. We assume first that the atomic thermal displacements are the combination of internal and rigid body motions[69],

$$\begin{aligned} \Delta X_i &= \Delta X_i^{rigid} + \Delta X_i^{int} \\ &= \mathbf{t} + \boldsymbol{\omega} \times X_i^0 + \Delta X_i^{int} \end{aligned} \tag{17}$$

where $\times$ indicate cross product, $\mathbf{t}$ is a translation vector and $\boldsymbol{\omega}$ represents a rotation. We assume also that rigid-body motions and internal motions are independent of each other,

$$< |\Delta X_i|^2 > = < |\Delta X_i|^2 >^{rigid} + < |\Delta X_i|^2 >^{int} \tag{18}$$

or, expressed as B-factors,

$$B_i^{calc} = B_i^{rigid} + B_i^{int} \tag{19}$$

As the same rotation and translation apply to all atoms, all the $B_i^{rigid}$ depend on 10 parameters (see below), while the $B_i^{int}$ depend on the flexibility constants associated with each atom. Calibrating the ENM therefore amounts to finding the values of those 10 parameters and of the flexibility constants that minimize

$$\chi^2 = \sum_{i=1}^{N} (B_i^{exp} - B_i^{calc})^2 \tag{20}$$

in the following, we look in more details at the contributions of rigid body and internal motions, i.e. their explicit contributions to $\chi^2$, as well as the gradients of the latter with respect to the corresponding parameters.

### 2.5.2 Rigid body motions

The contribution of rigid body motion is relatively straightforward[69,70],

$$\begin{aligned} B_i^{rigid} &= \frac{8\pi^2}{3} < |\Delta X_i|^2 >^{rigid} \\ &= \frac{8\pi^2}{3}(< |\mathbf{t}|^2 > +2(X_i^0, < \mathbf{t} \times \boldsymbol{\omega} >) + < (\boldsymbol{\omega} \times X_i^0, \boldsymbol{\omega} \times X_i^0) >) \end{aligned} \tag{21}$$

There are 10 parameters in this equation associated with $\mathbf{t}$ and $\boldsymbol{\omega}$, which we can write as $A = (a_0, a_1, \ldots, a_9)$,

$$\begin{aligned} B_i^{rigid} &= a_0 + a_1 X_{i1}^0 + a_2 X_{i2}^0 + a_3 X_{i3}^0 + a_4 X_{i1}^0 X_{i1}^0 + a_5 X_{i1}^0 X_{i2}^0 + a_6 X_{i1}^0 X_{i3}^0 + \\ & a_7 X_{i2}^0 X_{i2}^0 + a_8 X_{i2}^0 X_{i3}^0 + a_9 X_{i3}^0 X_{i3}^0 \end{aligned} \tag{22}$$

The derivatives of $B_i^{rigid}$ and therefore of $B_i^{calc}$ and $\chi^2$ with respect to the 10 parameters associated with rigid motions are straightforward from this equation.

### 2.5.3 Internal motions

The calculation of the mean-squared displacements in Eq. 16 necessitates to compute the inverse of the Hessian of that potential. In the case of the potential specified in Equation 1, the Hessian is singular; indeed, the quadratic potential $V$ only depends on interatomic distances and is therefore invariant with respect to translations and rotations. The null space of the Hessian $H$ is then of dimension at least 6, making H non invertible. The covariance matrix can still be calculated as the Moore-Penrose pseudo-inverse of $H$, which we note as $H^\dagger$. The computed B-factor associated with the internal motions predicted by ANM, $B_i^{int}$, is then

$$B_i^{int} = \frac{8\pi^2}{3} tr(H_{ii}^\dagger) \tag{23}$$

where $H^{\dagger}_{ii}$ is the $3 \times 3$ submatrix of $H^{\dagger}$ at position $H^{\dagger}(3i-2:3i, 3i-3:3i)$ in MATLAB notation.

We need expressions for the derivatives of $B_i^{int}$ with respect to the flexibility constants $k_j$. We note first that

$$\frac{dB_i^{int}}{dk_i} = \sum_{j|(ij)\in\mathcal{V}} \frac{k_j}{2k_{ij}} \frac{dB_i^{int}}{dk_{ij}} \tag{24}$$

Second, from equation 23 we see that all the $B_i^{int}$ are defined from the diagonal of the matrix $H^{\dagger}$, and as the derivatives of the diagonal of a matrix is the diagonal of the derivatives of that matrix, the derivatives of $B_i^{int}$ will be fully characterized from the derivatives of $H^{\dagger}$ with respect to the force constants $k_{ij}$. In equation 14, we expressed the derivatives of $H$ with respect to $k_{ij}$. The following proposition shows that the derivatives of $H$ and of $H^{\dagger}$ are directly related,

**Proposition 1.** *If all the force constants $k_{ij}$ are strictly positive,*

$$\frac{dH^{\dagger}}{dk_{ij}} = -H^{\dagger} \frac{dH}{dk_{ij}} H^{\dagger} \tag{25}$$

*Proof.* See appendix A. □

Replacing equation 14 into equation 25, we get

$$\frac{dH^{\dagger}}{dk_{ij}} = -(H^{\dagger}U_{ij}) \otimes (H^{\dagger}U_{ij}) \tag{26}$$

Let $N_c$ be the multiplicity of the zero eigenvalue of $H$. Then,

$$H^{\dagger}U_{ij} = \sum_{k=N_c+1}^{3N} \frac{(\mathbf{e}_k, U_{ij})}{\lambda_k} \mathbf{e}_k \tag{27}$$

where $\mathbf{e}$ and $\lambda$ are the eigenvectors and eigenvalues of $H$, respectively.

### 2.5.4 Optimization

The two previous subsections provide the full framework for computing the contributions of rigid motions and internal motions to the atomic position fluctuations, as well as the

derivatives of those fluctuations with respect to the parameters of the contributions, namely the 10 parameters $a_k$ for the rigid motions and the $N$ parameters $k_i$ for the internal motions. It is then possible to optimize those parameters so that the computed fluctuations match with the experimental B-factors by minimizing the $\chi^2$ given in equation 20. As the derivatives are known explicitly, we can use the BroydenFletcherGoldfarbShanno (BFGS) algorithm, a quasi-Newton method to perform this optimization. We use the L-BFGS-B variant of this algorithm[71], as it requires limited amount of memory and enables simple bound constraints on the variables that are optimized. This is important as we can then enforce positivity for the flexibility constraints and for $a_0$ that is expected to be positive.

# 3 METHODS

## 3.1 Data sets

To test the parameterization procedure described above, we used the dataset of proteins originally used by Xia et al[55] for a similar studies of fitting B-factors using NMA. This dataset contains 70 non-redundant proteins (see supplement S4 of Xia et al,[55]) whose structure has been solved by X-ray crystallography, with resolution better than 2.7 Å. These proteins vary in size from 40 amino acids to 298 amino acids.

Nine proteins were considered for comparing atomic position fluctuations observed in MD simulations and in the parameterized normal modes, three $\alpha$ proteins, 1AH7, 1LRV, 153L, three $\beta$ proteins, 1AQB, 1AG6, 1JPC, and three $\alpha + \beta$ proteins, 1A2P, 1AHQ, and 1PLR. These proteins vary in size from 100 to 259 residues. The MD trajectories were downloaded from the Molecular Dynamics Extended Library MODEL resource[72], available at `http://mmb.pcb.ub.es/MoDEL/`. All the MD simulations were performed using AMBER8.0[73], with param99 molecular force field and tip3P water model[74]. These simulations were performed on the monomeric protein, over 10 ns. More details on the simulations can be found at the MODEL web page.

## 3.2 Metrics for comparing experimental and computed B-factors

We use both correlation coefficients (CC) and root-mean square deviations (RMSD) as metrics for comparing B-factors. The CC are computed as Pearson's correlation coefficients,

$$CC = \frac{\sum_{i=1}^{N} \left(B_i^{exp} - \widehat{B^{exp}}\right)\left(B_i^{calc} - \widehat{B^{calc}}\right)}{\sqrt{\sum_{i=1}^{N} \left(B_i^{exp} - \widehat{B^{exp}}\right)^2 \sum_{i=1}^{N} \left(B_i^{calc} - \widehat{B^{calc}}\right)^2}} \tag{28}$$

where $B_i^{exp}$ and $B_i^{calc}$ are the experimental and computed B-factors for atom $i$, respectively, and $\widehat{B^{exp}}$ and $\widehat{B^{calc}}$ are the corresponding averages over the $N$ atoms considered.

The RMSD is defined as

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}(B_i^{exp} - B_i^{calc})^2}{N}} \tag{29}$$

Note that the optimization procedure described in the section 2 is set to minimize this RMSD between experimental and computed B-factors.

## 3.3 Atomic fluctuations from MD simulations

The covariance matrix $C_{MD}$ of atomic fluctuations during an MD simulation is derived from the snapshots along the trajectory as a sample statistic,

$$C_{MD} = \frac{1}{M-1} \sum_{m=1}^{M} (\mathbf{X}_m - \widehat{\mathbf{X}}) \otimes (\mathbf{X}_m - \widehat{\mathbf{X}}) \tag{30}$$

where $\mathbf{X}$ is the vector of dimension $3N$ specifying the coordinates of the atoms of the molecule, $\mathbf{X}_m$ is the value of that vector at the conformation $m$ in the trajectory which has been rotated and translated to minimize its cRMS to the experimental structure, $M$ is the total number of conformations in the trajectory, and $\widehat{\mathbf{X}}$ is the mean conformation over the trajectory

$$\widehat{\mathbf{X}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{X}_m \tag{31}$$

## 3.4 Comparing MD simulations and NMA

Both NMA and MD capture the dynamics of a molecule. This dynamic can be represented with atomic fluctuations and the covariance of those atomic fluctuations, computed as a

covariance matrix. To assess how well ENM and MD match, we assume that the trajectories they generate follow multivariate normal distributions. This assumption is justified for ENM, and only approximate for MD simulations. Referring to those distributions as $D_{ENM} = \mathcal{N}(\mu_{ENM}, C_{ENM})$ and $D_{MD} = \mathcal{N}(\mu_{MD}, C_{MD})$ for ENM and MD, respectively, where $\mu$ is the mean conformation and $C$ the covariance matrix, we use the Bhattacharyya distance[75] to evaluate their similarities,

$$D_B(D_{ENM}, D_{MD}) = \frac{1}{8}(\mu_{ENM} - \mu_{MD})^T C^{-1}(\mu_{ENM} - \mu_{MD}) + \frac{1}{2}\ln\frac{\det C}{\sqrt{\det C_{ENM}\det C_{MD}}} \tag{32}$$

where $C = \frac{C_{ENM}+C_{MD}}{2}$. Note that in this expression of the Bhattacharrya distance, the first term is related to the Mahalanobis distance, while the second term is related to the Jensen-Bregman LogDet divergence[76]. Computation of the latter term requires caution, as the covariance matrices are not full rank (due to their invariance with respect to rigid motions) and therefore their determinants are zero. We apply the rank normalization introduced by Fuglebakk et al[57,77] to correct for this rank deficiency.

For convenience we will report the similarity as the Bhattacharyya coefficient, $BC$,

$$BC((D_{ENM}, D_{MD}) = e^{-D_B(D_{ENM}, D_{MD})}. \tag{33}$$

This coefficient is between 0 and 1, with 0 indicating poor similarity, and 1 indicating perfect match, reached when the two distributions are identical.

## 3.5 Overlaps between normal modes and structure displacements

Let us consider a molecular system $S$ with $N$ atoms for which we have two conformations, **A** and **B**. The conformational change between those two conformations is captured by a displacement vector, **D**, such that $\mathbf{D} = \mathbf{B} - \mathbf{A}$.

Let us now consider a set of $k$ normal modes for $S$ in conformation **A**. These normal modes have been computed based on the eigenvalues $\lambda$ and eigenvectors **e** of the Hessian of an elastic network for $A$. Under the normal mode model, the dynamics of **A** can be described as a linear superposition of the fundamental motions described by those eigenvectors. The corresponding dynamic that will bring **A** closer to **B** is obtained by assigning the weights

$W$ of the modes in this superposition through projections of the displacement vector onto the eigenvectors:

$$W = E^t D \tag{34}$$

where $E$ is the matrix of eigenvectors. The contribution of mode $i$ to this optimal collective change of conformation can then be measured as the absolute value of the cosine of the angle between the displacement, and the direction of the mode, given by its eigenvector $\mathbf{e}_i$:

$$O_i = \frac{|\langle \mathbf{e}_i, \mathbf{D} \rangle|}{||\mathbf{e}_i|| ||\mathbf{D}||} \tag{35}$$

$O_i$ takes values between 0 and 1, with small values indicating that the mode $i$ contribute little to the conformational change, while large values indicate a significant contribution. We note that $\sum_{i=1}^{3N} O_i^2 = 1$, as the $\mathbf{e}_i$ are normalized to 1 and are orthogonal to each other. Then, $SO_k = \sum_{i=1}^{k} O_i^2$ is a measure of the contribution of the first $k$ normal modes to the total overlaps between the normal modes of **A** and the displacement between **A** and **B**. Note that when $k = 3N$, $SO_k = 1$.

## 3.6 Packing density

Following Halle[4], the local packing density $n_i$ of an atom $i$ in a protein can be computed from the X-ray structure by first defining a radial distribution function $g_i(r)$ as (see equation 6 in Ref.[4]):

$$g_i(r) = \frac{1}{4\pi r} \sum_j \frac{e^{-\frac{(r-r_{ij}^0)^2}{2\sigma_j}} - e^{-\frac{(r+r_{ij}^0)^2}{2\sigma_j}}}{\sqrt{2\pi\sigma_j} r_{ij}^0} \tag{36}$$

where $\sigma_j$ is the mean-square displacement of atom $j$, $r_{ij}^0$ is the distance between atom $j$ and atom $i$ in the X-ray structure, and the sum extends over all non-hydrogen atoms $j$ that are within a distance $R_c$ of $i$. The contact density $n_i$ is then given by:

$$\begin{aligned} n_i &= \int_0^{R_c} 4\pi r^2 g_i(r) dr \\ &= \sum_j \left( \frac{\sqrt{\sigma_j}}{\sqrt{2\pi} r_{ij}^0} \left( e^{-\frac{(R_c+r_{ij}^0)^2}{2\sigma_j}} - e^{-\frac{(R_c-r_{ij}^0)^2}{2\sigma_j}} \right) + \frac{1}{2} \operatorname{erf} \frac{(R_c + r_{ij}^0)}{\sqrt{2\sigma_j}} + \frac{1}{2} \operatorname{erf} \frac{(R_c - r_{ij}^0)}{\sqrt{2\sigma_j}} \right) \end{aligned} \tag{37}$$

# 4 RESULTS AND DISCUSSION

Coarse-grained normal mode analysis popularized by M. Tirion[31] are based on a simple elastic potential that is quadratic, with the crystal structure at its minimum, and defined over a geometric structure computed over the molecule of interest, the elastic network model (ENM). Here we focus on the construction of this ENM and its parameterization using experimental B-factors, as well as on the validity of such parameterization. In all computer experiments, NMA were performed based on a coarse-grained representation of the proteins that only consider the $C_\alpha$ atom of each of its residues. We note that other coarse grained models are available[25]; the CA-only model is the most common model used for coarse-grained normal mode analyses. All NMA computations are performed using our own program, FitNMA, written in C++. The source code for FitNMA is available at `https://www.cs.ucdavis.edu/~koehl/Projects/index.html`.

## 4.1 Building and parameterizing Elastic Network Models

We tested three different types of geometric ENMs, one based on the Delaunay triangulation of the positions of the $C_\alpha$ atoms in the protein of interest, and the two other based on a distance cutoff $R_c$. In the first ENM, referred to as DEL, a pair of $C_\alpha$ atoms is included if it forms an edge of the Delaunay triangulation, while in the two others, the same pair is included if the distance between their positions is smaller than $R_c$. We considered two values for $R_c$, i.e. 14 Å, which is within the range of values (13-15) usually considered for $C_\alpha$-based ENMs[51], and a larger cutoff of 20 Å. The corresponding ENM are referred to as EL14 and EL20, respectively. Each ENM was then parameterized with respect to the experimental isotropic B-factors of the $C_\alpha$ from the crystal structure, using the procedure described in section 2. Briefly, $C_\alpha$ $i$ of the protein is assigned a flexibility constant $k_i$. The link between two $C_\alpha$ $i$ and $j$ in the ENM is then assigned a force constant $k_{ij}$ that is the geometric mean of the flexibility constants of $i$ and $j$. Normal modes are computed based on the corresponding ENM, and the corresponding atomic fluctuations are compared to the experimental B-factors, taking into account possible rigid motions. The flexibility constants and the parameters associated to the rigid motions are then adjusted until the experimental

and computed atomic fluctuations match, in the least square sense.

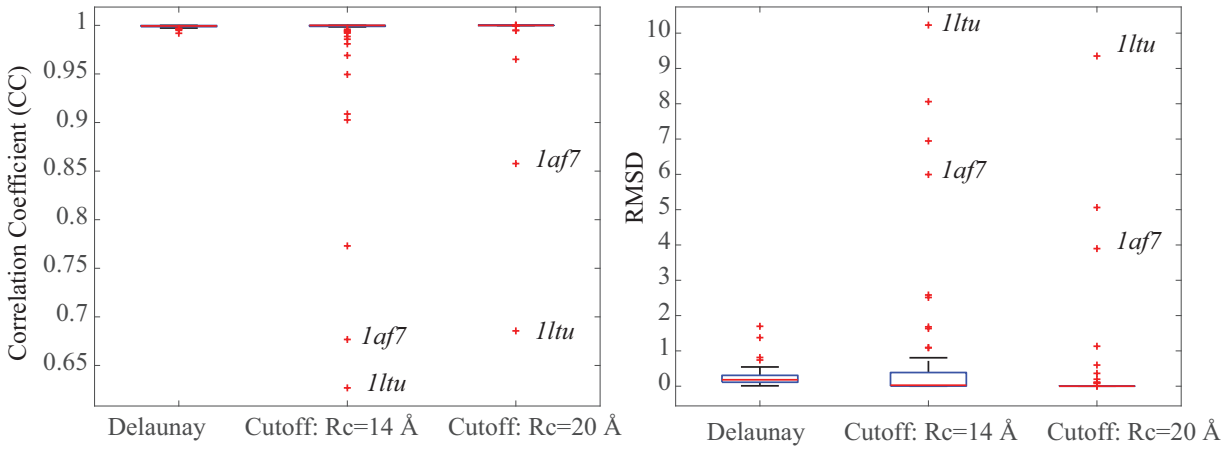



Figure 2: Boxplots of the correlation coefficients (left) and of the RMSD (right) between the computed B-factors and the experimental B-factors over the set of 69 proteins in our dataset, for the three types of ENM, based on Delaunay triangulation, based on a cutoff $R_c = 14$ Å, and based on a larger cutoff $R_c = 20$ Å. While the fits are relatively consistent for all three types of ENM, note the presence of a few outliers for the ENMs based on cutoffs. The two main outliers, 1LTU and 1AF7 are identified (see text for details).

We performed the analysis on a set of 70 high resolution protein structures (see Methods). The proteins included in this set are diverse, with sizes varying from 40 amino acids to 298 amino acids. In Figure 2, we compare the distributions of correlation coefficients CC and RMSD between experimental and compute B-factors at convergence of the fitting procedure, for all three types of ENMs. Overall, the fits are nearly perfect for all three types of ENM, with the average values for CC over the set of proteins are 0.999, 0.98, and 0.993 for DEL, EL14, and EL20, respectively, and the corresponding average RMSD values are 0.27 Å$^2$, 0.70 Å$^2$, and 0.30 Å$^2$, respectively. There are however a few outliers for the two ENMs based on cutoffs, for which the fitting procedure fails. We focus here on the two most significant ones for the computation based on a cutoff of 14 Å namely the apo structure (i.e. no iron) of a phenylalanine hydrolase of *chromobacterium violaceum* (PDB code 1LTU), and a methyltransferase from *salmonella typhimurium* (PDB code 1AF7) (see Figure 2). The structures of those two proteins and the corresponding ENMs are illustrated in Figure 3.

It is known that ENMs based on cutoff values are capable of reproducing experimental B-

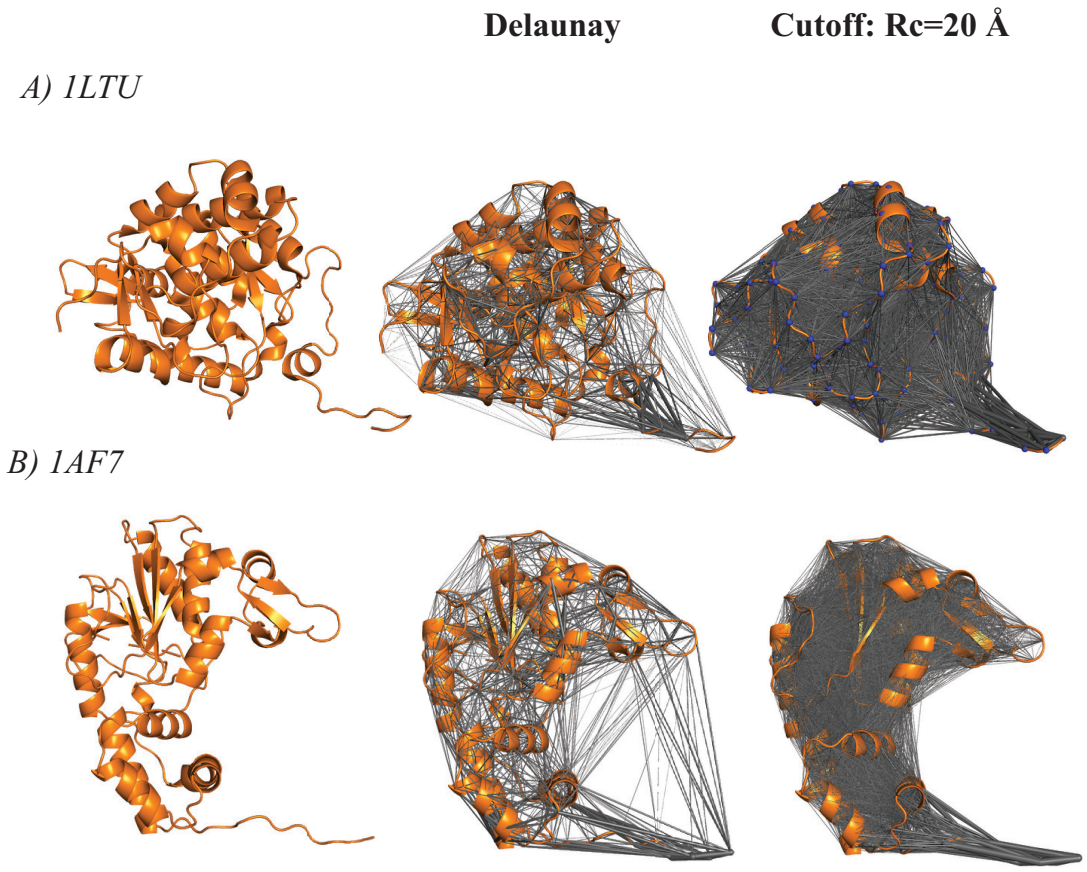



Figure 3: The optimized ENMs of a phenylalanine hydrolase (PDB code 1LTU), top, and of a methlytransferase (PDB code 1AF7). These proteins illustrate differences between the Delaunay-based ENM, and the cutoff-based ENMs, as for both of them the parameterization of the ENMs based on experimental B-factors failed for the cutoff-based ENMs. From left to right: Cartoon representation of the protein, elastic network (gray bonds) based on a Delaunay construct, and elastic network (gray bonds) based on a cutoff of 20 Å. The Delaunay networks contain 2030 and 1934 edges for 1LTU and 1AF7, respectively, while the corresponding cutoff-based networks with $R_c = 20$ Å contain 13809 and 12141 edges, respectively.

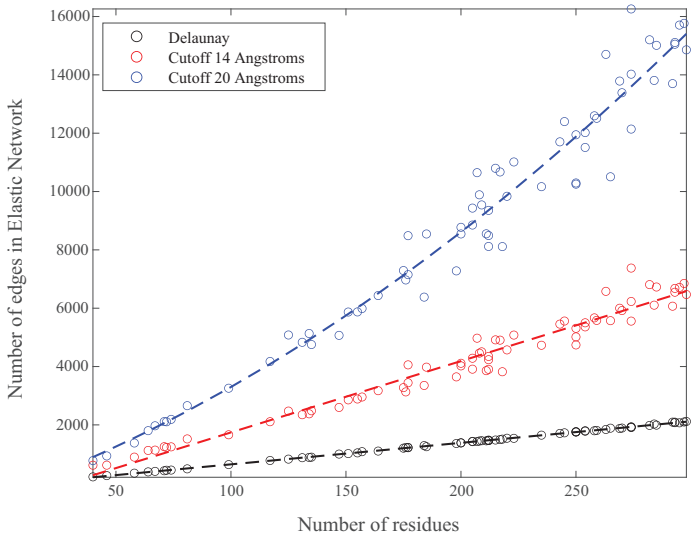



Figure 4: Size (i.e. number of edges) for the three types of ENM considered, i.e. based on Delaunay triangulation, DEL, based on a cutoff $R_c = 14$ Å, EL14, and based on a larger cutoff $R_c = 20$ Å, EL20, for all 70 proteins in our dataset. Dashed lines represent linear fit to the data for DEL and EL14, and a quadratic fit to the data for EL20. The corresponding $R^2$ are 0.99, 0.98, and 0.98, respectively.

factors well for globular proteins[51]. Indeed, such ENMs capture well their packing densities which play a dominant role in their dynamics. In contrast, it has been observed that such cutoff-based ENMs often fail for protein with an irregular shape[55]. We observe the same behavior here with the two proteins 1LTU and 1AF7 (1LTU was already identified as an outlier[55]). Both include a long flexible segment at their N-terminal region. Using a cutoff distance of 14 Å or even 20 Å, the cutoff-based ENMs only follow locally those long segments, while the Delaunay-based ENM provides a better connection of those segments with the rest of the proteins, thereby allowing for a better representation of their dynamics, as observed when fitting the B-factors. The same observations apply to the other outliers (results not shown).

There is another advantage in using a Delaunay-based ENM rather than a cutoff based ENM, as illustrated in Figure 4. The DEL ENM contains a significantly smaller number of edges than the EL14 and EL20 ENMs (on average a factor of 3 and 6.2 less, respectively), while still capturing the geometry of the molecule, as vouched by its ability to reproduce

experimental B-factors (see above). We believe that this is due to the fact that cutoff-based ENMs contain a lot of redundant information, while by construction Delaunay edges are more independent. This was already observed for distance-based statistical potentials for proteins[78].

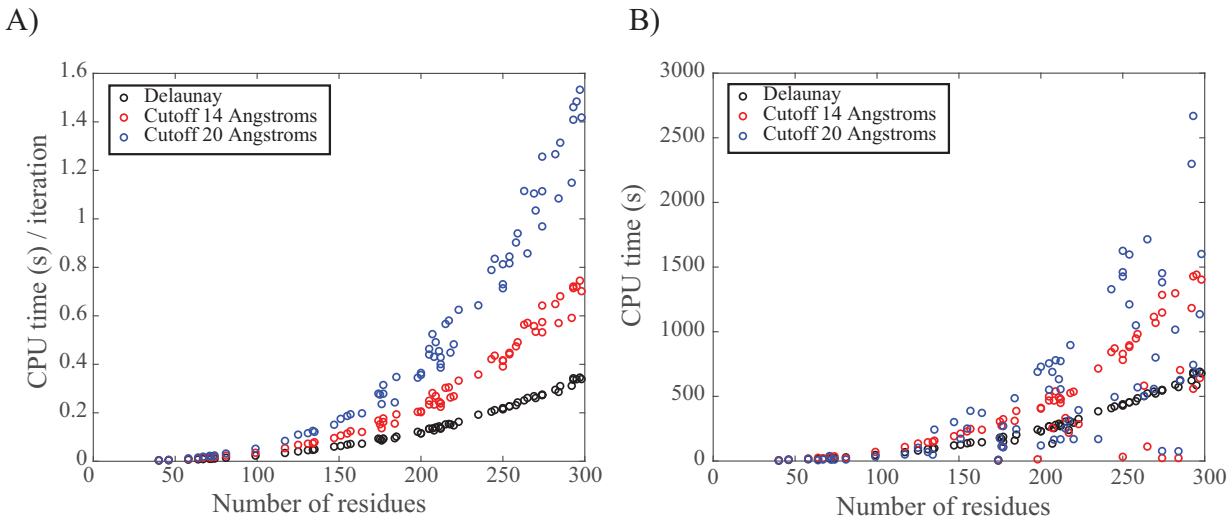



Figure 5: Parameterization of the EN is the result of a non-linear optimization process. We show the mean computing time per iteration in this optimization (panel A) and the total computing time (panel B) for the three types of ENM considered, i.e. based on Delaunay triangulation, DEL, based on a cutoff $R_c = 14$ Å, EL14, and based on a larger cutoff $R_c = 20$ Å, EL20, for all proteins in our dataset. All computations were performed on an Intel Core i7 processor with 8 cores running at 4.00GHz, and 64GB of memory.

The results presented above hint at using the Delaunay-based ENM to capture correctly the geometry of a protein structure. To parameterize this Delaunay-based ENM, we have used the procedure described in section 2. In this procedure, the computed B-factors, as well as their derivatives with respect to the atomic flexibility constants are all based on the Moore-Penrose pseudo inverse $H^\dagger$ of the Hessian $H$ of the quadratic potential, see equations 23 and 27. This pseudo inverse is computed over all non-zero eigenvalues of $H$ and their corresponding eigenvectors. Including all those eigenvalues comes at a computational cost as illustrated in figure 5.

Our model includes the contributions of rigid motions and internal motions when com-

puting atomic position fluctuations. It is based on 10 parameters for the rigid motions and $N$ parameters, the atomic flexibilities $k_i$, for the internal motions. Those parameters are optimized such that the computed fluctuations match with the experimental B-factors. This is a non-linear optimization, which we solve using an iterative BFGS procedure (see Methodology above). Each iteration involves computing the Moore-Penrose pseudo inverse of the Hessian matrix $H$, which is obtained from the eigen-decomposition of $H$, as well as its derivatives with respect to the atomic flexibilities $k_i$. We used the LAPACK routine dsyev to perform the eigen-decomposition. Dsyev assumes that the matrix $H$ is dense; as such, this computation depends on the number of atoms, and not the size of the EN. The situation is different for the derivatives. From proposition 1 and equations 24, computing those derivatives scales linearly with the number of edges in the EN. In figure 5A, we do observe the impact of the size of the EN on the computing per iteration of the non linear optimization, as Delaunay-based EN that contain significantly less edges lead to much shorter computing time. The same effect is observed for the overall computing time (figure 5B), but with some outliers. Indeed, some parameterizations of large cutoff based EN can be less demanding in computing time, as those parameterizations require less iterations. On average, each optimization requires 2000 iterations (with convergence defined with the norm of the derivative vectors is below $10^{-4}$).

The overall computing cost of parameterizing the EN of a protein is large: it takes on average 300 s on an Intel Core i7 processor with 8 cores running at 4.00GHz for the Delaunay-based EN, and 2500s for the cutoff-based EN. A significant fraction of the cost comes from the full diagonalization of the Hessian matrix at each iteration of the optimization of the parameters. We tested if it is possible to only include a fraction of the eigenpairs of the Hessian matrix, those corresponding to the smallest eigenvalues that are related to the largest collective internal motions[31]. Results are shown in figure 6, for the Delaunay-based ENM. Similar results are observed for the cutoff-based ENMs (results not shown). We note however that using only a fraction of the eigenpairs of the Hessian $H$ when computing its Moore-Penrose pseudo-inverse $H^{\dagger}$ and its derivatives is a major approximation that significantly reduce the performance of the parameterization of the ENM. While the performance increases (i.e. increased CC and reduced RMSD) as the number of modes increases, it remains that

good parameterization is only observed when all modes are included.

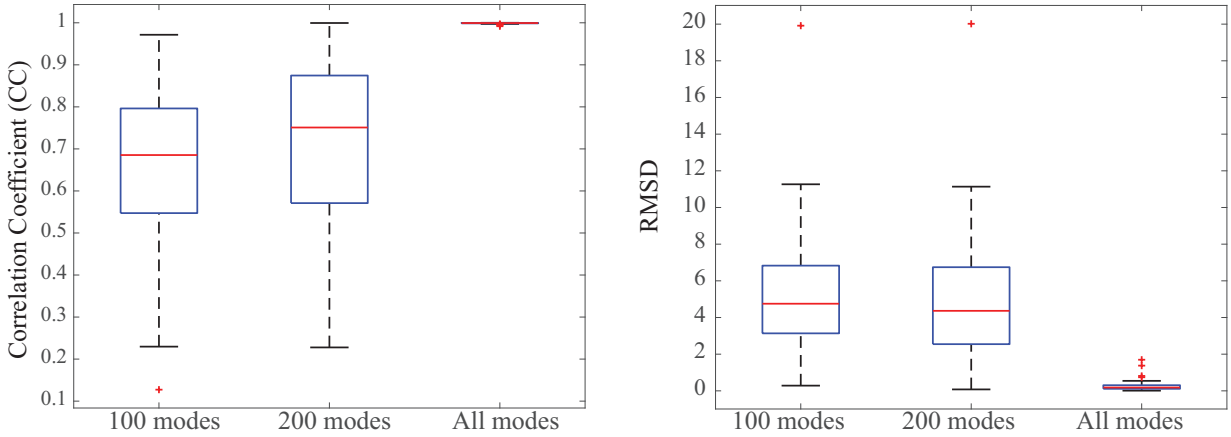



Figure 6: Boxplots of the correlation coefficients (left) and of the RMSD (right) between the computed B-factors and the experimental B-factors over the set of 70 proteins in our dataset, for the Delaunay-based ENM, with different numbers of eigenpairs of the Hessian $H$ included in the computation of the Moore-Penrose pseudo-inverse $H^\dagger$ and its derivatives.

## 4.2 Appraising the parameterized ENMs

Validation of normal mode analysis based on coarse grained ENMs is usually performed by comparing the atomic fluctuations induced by those normal modes with the crystallographic B-factors. Such a comparison is futile in our setting, as the ENMs have been parameterize such that they reproduce those experimental B-factors (nearly) exactly. We rely instead on comparison with MD simulations, as well as by measuring how well the parameterized normal modes can capture conformational changes.

Coarse-grained NMA and MD simulations are two techniques that simulate the dynamics of a molecule computationally. While the former is based on a simplified geometric model of the protein (the ENM) and a simplified quadratic potential, the latter are based on usually detailed, anharmonic potentials that have been parametrized semi-empirically (note that coarse-grained MD simulations have been developed, see for example M. Levitt[79]). As MD simulations are usually more detailed and often considered to reproduce correctly experimental results, many have resulted in benchmarking different ENM models for NMA

Table 1: Bhattacharyya Coefficients Comparing MD Covariances with Covariances Predicted from NMA with Different ENMs

| Class | PDB ID | Nres | DEL-1 [a] | DEL-opt [a] | EL20-1 [a] | EL20-opt [a] |
|---|---|---|---|---|---|---|
| $\alpha$ | 1AH7 | 245 | 0.87 | **0.88** | 0.75 | 0.77 |
| $\alpha$ | 1LRV | 233 | 0.61 | **0.74** | 0.46 | 0.74 |
| $\alpha$ | 153L | 185 | 0.89 | **0.90** | 0.67 | 0.88 |
| $\beta$ | 1AQB | 175 | 0.85 | **0.87** | 0.73 | 0.81 |
| $\beta$ | 1AG6 | 100 | 0.80 | **0.90** | 0.77 | 0.82 |
| $\beta$ | 1JPC | 109 | 0.84 | **0.85** | 0.64 | 0.83 |
| $\alpha+\beta$ | 1A2P | 109 | 0.87 | 0.90 | 0.77 | **0.91** |
| $\alpha+\beta$ | 1AHQ | 134 | 0.90 | **0.91** | 0.66 | 0.90 |
| $\alpha+\beta$ | 1PLR | 259 | 0.84 | **0.87** | 0.56 | 0.86 |

[a)] 1 indicates that all the edges of the ENM were assigned the same force constant, 1, while "opt" indicates instead that the ENM was parameterized using the experimental B-factors

[b)] The highest coefficients are highlighted in bold. Note that the largest the coefficient, the more similar the covariance matrices from MD and from NMA are.

against MD (see Ref.[57,80–83], among others). We repeat their analyses here to benchmark our parameterized ENMs.

We used a dataset of 9 proteins, three from each structural class (mainly $\alpha$, mainly $\beta$, and $\alpha+\beta$). For all those structures, we use MD simulations previously published and available at the Molecular Dynamics Extended Library MODEL resource[72]. All those simulations were performed using AMBER, with the param99 forcefield and the tip3p water model. Most of those simulations were performed over 10ns, with the exception of Lyzozyme (PDB code 153L), with a total simulation time of 100 ns, and barnase (PDB code 1A2P), with a simulation time of 13.5 ns. For all simulations, we superimposed all frames in their trajectory to the PDB structure to remove rigid motions. We then computed a mean structure, and the covariance of the atomic fluctuations, as described in the Method section. For the NMA analyses, we generated the Delaunay-based ENM and the cutoff-based ENM (with $R_c = 20$ Å) starting from the mean MD structure, and parameterized those ENMs using the

experimental B factors from the PDB structure. The covariance matrices are then simply the Moore-Penrose pseudo inverses of the Hessian matrices of the parameterized energy of the ENM. As both NMA and MD simulations capture dynamics as variations around the mean MD structure, the Bhattacharyya distance between their distributions of conformations is reduced to the Jensen Bregman LogDet divergence that directly measures the similarities between the covariance matrices of the distributions (see Method). The similarities are reported as Bhattacharyya coefficients $BC$ that vary between 0 and 1, with 0 indicating no similarity, and 1 perfect similarity. Results for the 9 proteins are shown in table 1.

In their evaluations of ENMs using a comparison with MD simulations, Fuglebakk et al.[57] stated that "It is however not clear that agreement between atomic fluctuations of models imply agreement between their covariance structures", to finally reach the conclusion that "the ENM models that agree best with B-factors model collective motions less reliably and recommend against using B-factors as a benchmark". Here we show in contrast to these findings that parameterization of the ENM using the experimental B-factors improve the similarity of the covariance matrices computed from MD and computed from the ENM. As seen in table 1, the improvement is often small but systematic, and can be large, such as the plastocyanin from spinach (PDB code 1AG6), a compact small $\beta$ protein. Interestingly, the improvement is always more significant for the cutoff-based ENM. The covariance of the parameterized Delaunay-based ENM remains, however, more similar to the covariance of the MD simulations than the covariance of the cutoff-based ENMs, with one exception, barnase (PDB code 1A2P).

## 4.3 Capturing conformational changes with normal modes

One of the main applications of coarse-grained NMA based on ENM is to study functional conformational changes. By studying proteins for which multiple structures have been resolved in different conformations (such as open and closed states, apo and holo forms with respect to a ligand), it has been shown that the low frequency normal modes of the ENMs correlate well with the functional conformational changes[32,50,84–86] It is this somewhat surprising observation (as ENM computations are only valid for very small deviations around the equilibrium) that has popularized coarse-grained NMAs based on ENMs. Here we assess

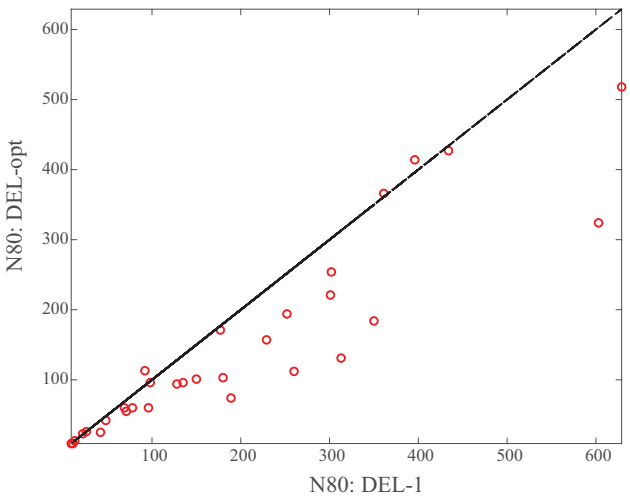



Figure 7: The N80s, i.e. the number of modes needed such that the overlap between normal modes and conformational changes reaches 80% are compared for normal modes based on constant force Delaunay-based ENM (DEL-1) and based on a Delaunay-based ENM parameterized with experimental B-factors (DEL-opt). The dotted line represents the first diagonal. Points below the diagonal indicate that less normal modes computed from the DEL-opt ENM are needed to represent conformational changes.

if parameterizing the ENM using the experimental B-factors help when attempting to capture conformational changes in proteins using a small number of normal modes. We used a data set of 31 pairs of protein structures originally designed by Bastolla and Dehouck[86]. The list of protein pairs can be found in table S1 of the supporting information of their paper[86]. Each pair corresponds to two distinct structures of the same protein chain, representing a conformational change that is relevant for its function. The coordinate root-mean-square deviation (cRMS) between structural pairs ranges from 0.35 to 34.4 Å. One structure in each pair is considered as the initial conformation. For each protein, we build its Delaunay-based ENM and consider two versions of this ENM, one in which all edges are assigned a force constant of 1, DEL-1, and one in which the force constants are parameterized using the experimental B-factors using the procedure described above, DEL-opt. We then assess how the modes associated with these ENMs can be used to map the conformational changes of the structures. We use the overlap between the modes and the conformational displacement to assess this mapping. The overlap is cumulative with respect to the number of modes that

are considered (see Method). We then estimate the number of modes $N80$ that is needed to reach 80% overlap between the normal modes and the conformational changes. The numbers $N80$ obtained for DEL-1 and DEL-opt are compared in Figure 7.

In most cases, the number of normal modes needed to represent the conformational changes for the proteins considered is less for the parameterized ENM than for the constant ENM. This result, while supporting the rationale for parameterizing ENMs using experimental information on dynamics, should still be considered with caution at this stage, as it is provided for illustration here. This analysis should be repeated on a much larger number of proteins.

## 4.4 Amino acid flexibility constants

The procedure described in this paper performs a parameterization of the force constants associated with the edges of the ENM describing the protein. Instead of refining directly those force constants, we express them as the geometric average of the flexibility constants of the residues that form those edges. From a computational perspective, this has the advantage of reducing significantly the number of degrees of freedom in the optimization process from $O(N^2)$ to $O(N)$, which is of significance as the experimental information used for the parameterization is of order $O(N)$. Introducing more degrees of freedom than constraints would significantly increase the risk of overfitting. The question now is to see if there is some meaning to the actual parameters that are refined, namely the constants $k_i$ for residues $i$, which we have dubbed as 'flexibility constants". To better understand those parameters, we have analyzed their values for all residues in our dataset of proteins, with the exception of 1AMM, and compared them with similar analyses of the corresponding B-factors, which are much better understood. Note that we have removed the protein with PDB code 1AMM, i.e the bovine eye lens protein gamma B Crystallin, as its structure was determined at 150K; as such, its B-factors are significantly lower and cannot be compared directly with those of proteins whose structures were studied at a higher temperature. We have used the values derived from the parameterization of the Delaunay-based ENMs of those proteins. We also computed the accessible surface areas (ASA) of all residues in those proteins, using the procedure introduced by Le Grand and Merz[87]. We report the results of those analyses per

amino acid type. We use the median as a statistics, as the underlying distributions are not symmetric (see for example Vihinen et al[88] for illustrations of the distributions of B-factors). Note that we did not normalize the values of B-factors, flexibility constants, and ASA, as originally suggested by Karplus and Schulz[89]; while we agree that there might be biases in those values, we are more interested in qualitative average behaviors. Results of our analyses are presented in figure 8.

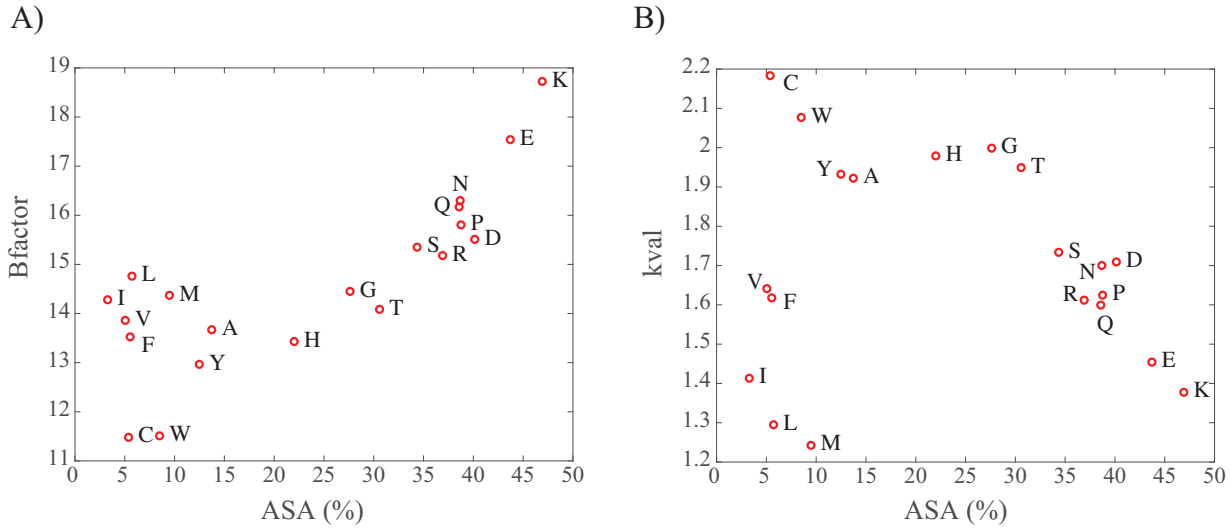



Figure 8: The median Bfactors (subplot A), and the median flexibility constants (subplot B) for all amino acid types are plotted against their median accessible surface areas. The distributions of B-factors, flexibility constants, and ASA are computed over all 69 proteins of our dataset. As those distributions are not symmetric, we consider the median instead of the mean.

B-factors as reported in protein crystal structures reflect the fluctuation of an atom about its average position. A large B-factor is usually indicative of high mobility of the corresponding residue, usually within its side chain. B-factors have been analyzed to define a flexibility scale for amino acids[88–90], which in turn can be used to predict protein flexibility, as well as disordered regions in proteins[91]. Intuitively it is expected that residues with high mobility are more accessible to the solvent[9]. This is indeed observed in our dataset of proteins, as illustrated in Figure 8A. The hydrophobic residues, whose median accessibilities are low, have low median B-factors. In contrast, the hydrophilic residues, especially the large

charged residues Lysine and Glutamate, are on average highly accessible and highly mobile, i.e. with large B-factors. Surprisingly, the flexibility constants exhibit an opposite behavior, as illustrated in Figure 8B. For most amino acids, there is a nearly linear relationship between the $k$ constants and ASAs, but with a negative slope, i.e. accessible residues have lower flexibility constants. The five hydrophobic residues V, I, L, M, and F, are exceptions to this relationship, as they have on average low accessibility and low flexibility constants.

## 4.5 Flexibility vs B-factors

Table 2: Indicators for predictions of C$\alpha$ B-factors

| | Model [a] | Predictor | $< CC >$ [b] | Range of CC |
|---|---|---|---|---|
| $a$ | LDM | Density $n_k$ | 0.57 ± 0.12 | 0.22 to 0.83 |
| $b$ | DEL-opt | Flexibility constant | 0.53 ± 0.14 | 0 to 0.8 |
| $c$ | DEL-opt | Frequency $\Omega$ | 0.65 ± 0.19 | 0 to 0.9 |
| $d$ | DEL-1 | Frequency $\Omega$ | 0.14 ± 0.09 | 0 to 0.48 |
| $e$ | EL20-opt | Flexibility constant | 0.43 ± 0.23 | -0.34 to 0.83 |
| $f$ | EL20-opt | Frequency $\Omega$ | 0.77 ± 0.16 | 0.16 to 0.97 |
| $g$ | EL20-1 | Frequency $\Omega$ | 0.54 ± 0.15 | 016 to 0.82 |

[a)] LDM is the local density model of Halle[4]. DEL and EL20 are elastic networks (EN) based on the Delaunay complex and a 20 Å cutoff, respectively, with 1 indicates that all the edges of the EN were assigned the same force constant, 1, while "opt" indicates instead that the EN was parameterized using the experimental B-factors

[b)] Correlation coefficients between the experimental B-factors and the inverse of the predictor values for all C$\alpha$. Results are given as mean value ± one standard deviation over the set of 70 proteins.

In a landmark paper, Halle[4] proposed that B-factors, or more specifically atomic mean square displacements (AMSDs), can be predicted solely on the basis of packing density. Subsequent studies have shown that the same idea applies to NMR, i.e. packing density is a predictor for NMR order parameters, $\mathcal{S}^2$[5,8]. In Halle's model, referred to as LDM for local

density model, each atom $i$ of a protein is characterized with an AMSD $\sigma_i$, which is related to the B-factor $B_i$ according to $B_i = 8\pi^2\sigma_i/3$. This AMSD is expected to be related to the local packing of $i$, defined based on its contact density, $n_i$, i.e. the number of (non-hydrogen) atoms within a spherical region centered on $i$. Namely,

$$\sigma_i = \frac{3}{2\lambda}\frac{1}{n_i} \tag{38}$$

where $\lambda$ is a scaling parameter that accounts for temperature. The atomic density $n_i$ (see equation 37) is itself a function of the $\sigma_k$ of the atoms $k$ in the neighborhood of $i$, i.e. a spherical region of size $R_c$. The $n_i$ and $\sigma_i$ are then computed self-consistently using equations 37 and 38, as described by Halle[4]. At convergence, the computed $\sigma_i$ are scaled such that their mean value over a protein is equal to the mean experimental AMSD over the same protein. Halle showed that the resulting scaled converged $\sigma_i$ reproduce accurately the corresponding B-factors on a set of 38 proteins. We repeated his calculations on our set of 70 proteins, using $R_c = 7.32$Å as suggested by Halle, and found similar results (see Table 2, row *a*), albeit with lower accuracy. The difference is most likely due to the fact that we did not account for crystal contacts in our calculations, while Halle did.

Do our atomic flexibility constants also relate to packing density, or, based on Halle's results, do they correlate well with B-factors? Our results seem to indicate that this is not the case, both for the Delaunay based EN and for the cutoff based EN at least on our dataset of 70 proteins (see rows *b* and *e* of table 2, respectively for the correlations to the B-factors). The corresponding mean correlation coefficients between packing density, $n_k$, and flexibility constants, $k_k$, are 0.06 ± 0.09 (with a range -0.27 to 0.28) for the Delaunay based EN, and -0.15 ± 0.14 (with a range -0.57 to 0.20), for the EL20 cutoff-based EN, i.e. poor correlations in both cases. These observations allow us to better understand the flexibility constants we have introduced. Unlike packing density that captures the local environment of an atom, the atomic flexibility constant is an intrinsic dynamic property of the atom itself. It is the local network, namely the list of edges in the EN that connect to an atom $k$ that defines the local environment of an atom. To test if this is the case, we have assigned to each atom $i$ a

frequency $\Omega_i$[92] such that

$$\Omega_i^2 = \sum_{j=1}^{N(i)} k_{ij} = \sum_{j=1}^{N(i)} \sqrt{k_j k_i} \tag{39}$$

where the summation extends over all atoms $j$ such that $ij$ is an edge in the EN considered. We computed $\Omega_i$ for all C$\alpha$ atoms of all proteins in our dataset, for the Delaunay based EN and the cutoff based EN, with constant, or optimized values for the force constants $k_{ij}$. Results are shown in table 2 on rows $c$ and $d$ for the Delaunay based EN, and on rows $f$ and $g$ for the cutoff-based EN. As expected, the inverse of the frequencies $\Omega$ correlate well with the B-factor values, indicating that these frequencies capture the impact of the local environment of an atom on its dynamics. We note that the frequencies computed from the optimized force constants show stronger correlations with the B-factors than the frequencies computed from constant force constants (rows $c$ vs $d$ and rows $f$ vs $g$). This may not be too surprising as the optimization is based on the B-factors.

## 4.6 Parameterized ENs capture rigidity

The differences between B-factors and flexibility constants suggest that the latter do not actually characterize residue mobility, as suggested in the name we gave them. We investigated the connection between flexibility constants and mobility within the broader framework of rigidity of proteins. Jacobs, Thorpe and collaborators pioneered the use of rigidity-based methods in protein flexibility analysis[63,64,93]. Their analysis is based on graph theory. They start by designing a constraint network on the protein of interest, much akin to the ENMs considered here, but with the significant difference that the constraint network is designed to capture the energetics of the protein, rather than its geometry. The constraint network includes all covalent bonds and strong hydrogen bonds within the protein of interest. They then run an algorithm, dubbed the 3D Pebble game, to count the degrees of freedom within this constraint network. From the listing of degrees freedom, the algorithm identifies 'all the rigid and flexible substructures in the protein, including over-constrained regions (with more bonds than are needed to rigidify the region) and under-constrained or flexible regions, in which internal motions can occur", paraphrasing the authors' descriptions of their

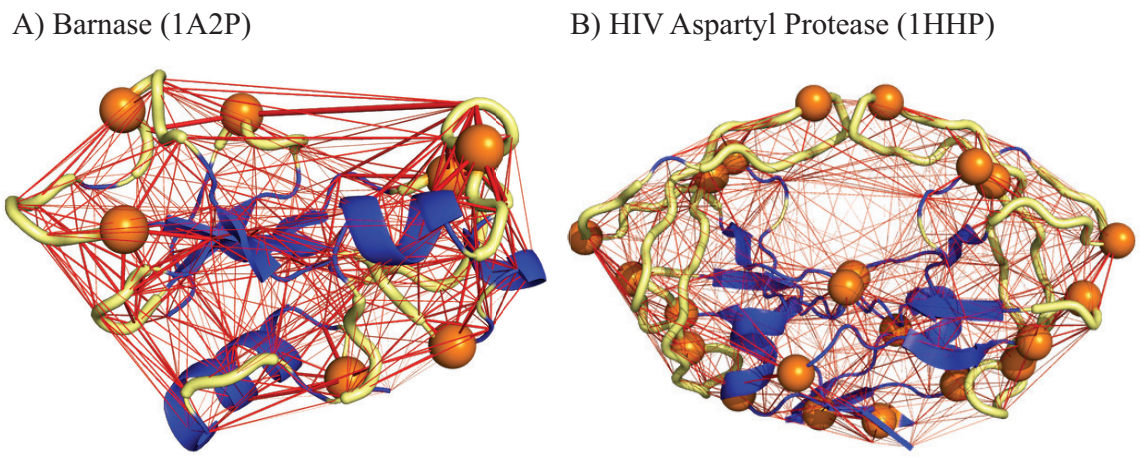



Figure 9: Rigidity analysis of Barnase (PDB code 1A2P), left, and of the ligand-free aspartyl protease of HIV-1 (PDB code 1HHP), right. The rigid and flexible regions of the proteins, as identified by the program ProFlex from the Leslie Kuhn group (see text for details) are shown in blue and pale yellow, respectively. The edges of the parameterized Delaunay-based ENM are shown in red. Residues with large flexibility constants are highlighted with orange spheres.

algorithm[63]. With this underlying definition of flexibility and rigidity, we mapped the positions of residues with large flexibility constants on the partitioning of a protein structure into rigid and flexible regions obtained with the 3D Pebble game algorithm for two proteins, barnase (PDB code 1A2P), and a ligand-free HIV protease (PDB code 1HHP). We use the implementation of the 3D Pebble game from the software package ProFlex developed by the group of Leslie Kuhn at Michigan State University, and available at the URL `https://kuhnlab.natsci.msu.edu/software/proflex/`. The residues with large flexibility constants were identified from the parameterized Delaunay-based ENMs for those two proteins. Results are shown in figure 9. Clearly, for those two proteins all residues with large flexibility constants fall within the regions defined as flexible by ProFlex. From the definition of flexibility in the theory behind ProFlex, those regions are under-constrained and then prone to internal motions. The parameterized ENMs implicitly capture this flexibility by assigning large force constants in those regions, with those large force constants allowing

for concerted motions as described by the normal modes of the ENMs. It is therefore more appropriate to refer to the parameterized constants $k_i$ as flexibility constants for the residues.

# 5 CONCLUDING REMARKS

Coarse-grained normal mode analyses rely on the idea that the geometry of a protein structure contains enough information for computing its fluctuations around its equilibrium conformation. This geometry is captured in the form of an elastic network, i.e. a network of edges between residues in the protein structure. A spring is then associated with each of these edges. The normal modes of the protein of interest are then identified with the normal modes of the corresponding elastic network. Constructing the elastic network and parameterizing this network remain topics of research and development in the computational biology community. In this paper, we advocate for using the edges of the Delaunay triangulation of the points representing the $C_\alpha$ atoms of the protein as the elastic network, and for parameterizing this Delaunay-based elastic network such that its dynamics match with the experimental B-factors of the $C_\alpha$ atoms. Both comes with some sacrifice in simplicity, but with benefits that we highlight below.

Computing a three dimensional Delaunay triangulation is more complex and more onerous in computing time than simply selecting the edges of an elastic network based on their lengths. A Delaunay-based elastic network, however, has several advantages, some of which are highlighted in figure 3. First, it is completely parameter-free: there is no need to define a cutoff value for selecting edges. Second, it leads to a much smaller elastic network in terms of number of edges. Finally, it is able to capture even long range contacts in the protein. The advantages of including long range interactions has been advocated before[57]. Other geometric constructions could replace the Delaunay triangulation, such as alpha shapes[65]. Such alpha shapes have already been considered for building elastic networks[55]. Finally, we note that many implementations of Delaunay triangulation algorithms are available, thereby mitigating the difficulties associated with their complexities.

An appealing aspect of coarse grained NMAs comes from the simplicity of their implementations. Besides the constructions of the elastic network described above, their parame-

terization is often simple, as a constant force constant is assigned to each edge, as prescribed by Tirion[31]. Here we advocate for parameterizing instead the elastic network such that its dynamic leads to atomic fluctuations that match with experimental B-factors. Most current models for fitting the EN to the B-factors are based on the assumption that the atomic displacements captured by B-factors result from internal motions of the protein structure. However, B-factors are known to be influenced by rigid-body motions taking place in the crystal[12]. In addition, contacts between molecules in the crystal are known to affect atomic fluctuations[58,94], such are other effects like twinning and lattice disorders[95]. We have chosen to focus on and include in our model the contribution of rigid-body motions, as several studies indicate that their effects are more important than those resulting from crystal contacts[60–62]. In addition, we were careful in reducing the risk of overfitting in this process by attaching a variable to each atom, and not to each edges in the network. We have shown that such a parameterization leads to improved NMAs, as it defines dynamics that is close to MD simulations (see table 1), as well as it reduces the number of normal modes needed to reproduce functional conformational changes (figure 7). In addition, the atomic constants defined in the parameterization process are found to be related to the concept of flexibility introduced in the protein rigidity theory introduced by Thorpe and co-workers (see figure 9).

The optimized normal mode model we propose is protein-specific, derived from the geometry of its static structure (in our study the X-ray structure), as well as from its dynamics as captured by the B-factors associated with the structure. Those B-factors, however, are indirect measures of dynamics and are subject to the refinement methods used to obtain them. There are options to circumvent this limitation. Diamond[96] and Kidera and Go[97] for example proposed independently to express the Debye-Waller factors directly in terms of normal modes, thereby allowing for atomic motions to be treated as anisotropic and concerted. In their models, the amplitudes (Diamond) or the amplitudes and directions (Kidera and Go) of those normal modes become parameters that are then refined against the experimental structure factors. Both models are derived from "standard" normal mode models, i.e. derived from a semi-empirical force-field. This idea was later expanded to the use of EN-based normal mode models (see for example Delarue and Dumas[43]). We see a potential extension

of our method in this direction. Instead of parameterizing the EN based on B-factors, we would use instead directly the experimental structural factors. Conversely, the Debye-Waller factors in the structure refinement would be written as functions of the force constants of the EN that models the dynamics of the protein, instead of the amplitudes and directions of their normal modes. We are currently exploring this extension of our model.

We reckon the increase in computational costs that comes with our procedure. We have expressed the parameterization of the elastic network as a non linear optimization problem whose parameters are the variables associated with rigid motions and the atomic flexibility constants associated with internal motions. While we are able to find analytical expressions both for the function that we minimize and for its derivatives, each iteration of the quasi Newton algorithm we use for the optimization is costly in computing time, as it requires that the Hessian of the quadratic potential of the elastic potential be diagonalized, and that all eigen pairs be computed. While this process can be parallelized, it remains a $O(N^3)$ process. We have tried to remove the requirement of using all eigen pairs, but found that this removal leads to loss of performance (figure 6). While the computation cost remains manageable for most protein structures available in the PDB (i.e. with up to 1000 residues), it can become an issue for larger protein complexes, such as viral envelopes. We are currently working on strategies for reducing significantly the computational cost of our procedure.

## ACKNOWLEDGMENTS

The work discussed here originated from a visit by P.K. at the Institut de Physique Théorique, CEA Saclay, France, during the fall of 2019. He thanks them for their hospitality and financial support.

## DATA AVAILABILITY STATEMENT

All structures of proteins used in this study are available in the public repository PDB (`www.rcsb.org`). We used a dataset of 70 non-redundant proteins, as identified in supplement S4 of Xia et al,[55]. Molecular dynamics trajectories of nine proteins (see text for details) were downloaded from the Molecular Dynamics Extended Library MODEL resource[72], available

at `http://mmb.pcb.ub.es/MoDEL/`.

# APPENDIX A

Let us first reintroduce some notations. $H$ is the Hessian of the quadratic potential defined in equation 1. We have shown[68] that H can be written as:

$$H = \sum_{(i,j)} k_{ij} \mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{40}$$

where the vector $\mathbf{U}_{ij}$ is defined as $\mathbf{U}_{ij} = (0, \ldots, 0, \frac{\mathbf{x_i} - \mathbf{x_j}}{r_{ij}}, 0, \ldots, 0, \frac{\mathbf{x_j} - \mathbf{x_i}}{r_{ij}}, 0, \ldots, 0)$. The derivatives of the Hessian with respect to any of the $k_{ij}$ are given by

$$\frac{dH}{dk_{ij}} = \mathbf{U}_{ij} \otimes \mathbf{U}_{ij} \tag{41}$$

As the fluctuations in atomic positions are related to the inverse $H^\dagger$ of the Hessian matrix $H$ (see equation 23), we also need the derivatives of this inverse. In the main text, we have stated the following proposition:

**Proposition.** *If all the force constants $k_{ij}$ are strictly positive,*

$$\frac{dH^\dagger}{dk_{ij}} = -H^\dagger \frac{dH}{dk_{ij}} H^\dagger \tag{42}$$

which we validate here.

*Proof.* The matrix $H$ can also be written as

$$H = \sum_{k=1}^{3N} \lambda_k \mathbf{e}_k \otimes \mathbf{e}_k \tag{43}$$

where $\lambda$ and $\mathbf{e}$ are the eigenvalues and eigenvectors of $H$, respectively. Note that that some of the $\lambda_k$ may be zero, i.e. the null space of H may not be empty. To account for this possibility, we prove the proposition separately in the case of an empty null space, and in the case of a null space with finite dimension.

## Case 1: All eigenvalues of $H$ are non zero

This is the easiest case and the proof of proposition 1 is simple. As all the eigenvalues are non zero, the matrix $H$ is invertible and its Moore Penrose inverse is its actual inverse. Therefore,

$$HH^{\dagger} = I \tag{44}$$

where $I$ is the $3N \times 3N$ identity matrix. Deriving this equation by $k_{ij}$, we get,

$$\frac{dH}{dk_ij}H^{\dagger} + H\frac{dH^{\dagger}}{dk_{ij}} = 0 \tag{45}$$

therefore,

$$\frac{dH^{\dagger}}{dk_{ij}} = -H^{\dagger}\frac{dH}{dk_ij}H^{\dagger} \tag{46}$$

which concludes the proof of proposition 1 for this specific case. Note that this case will not occur if the Hessian is based on Cartesian coordinates; it will occur, however, if the potential is computed based on internal degrees of freedom.

## Case 2: Some eigenvalues of $H$ are zero

As mentioned above, this is the general case when the potential and its Hessian are based on Cartesian coordinates. Indeed, as the potential is only function of interatomic distances, it is invariant with respect to rotations and translations, and therefore its Hessian will have (at least) 6 zero eigenvalues. For generality, we will define as $N_c$ the multiplicity of the eigenvalue 0 of H. The pseudo inverse of $H$ is then given by:

$$H^{\dagger} = \sum_{k=N_c+1}^{3N} \frac{1}{\lambda_k}\mathbf{e}_k \otimes \mathbf{e}_k \tag{47}$$

When $H$ is not full rank, equation 44 does not hold anymore. Indeed,

$$\begin{aligned} HH^{\dagger} &= \sum_{k=1}^{3N}\sum_{l=N_c+1}^{3N} \frac{\lambda_k}{\lambda_l}(\mathbf{e}_k \otimes \mathbf{e}_k)(\mathbf{e}_l \otimes \mathbf{e}_l) \\ &= \sum_{k=1}^{3N}\sum_{l=N_c+1}^{3N} \frac{\lambda_k}{\lambda_l}(\mathbf{e}_k, \mathbf{e}_l)\mathbf{e}_k \otimes \mathbf{e}_l \\ &= \sum_{k=N_c+1}^{3N} \mathbf{e}_k \otimes \mathbf{e}_k \end{aligned} \tag{48}$$

where the last equality comes from the fact that the eigenvalues $\mathbf{e}$ form an orthonormal base. We rewrite this equation as

$$HH^\dagger = I - \sum_{k=1}^{N_c} \mathbf{e}_k \otimes \mathbf{e}_k \tag{49}$$

where $I$ is the $3N \times 3N$ identity matrix. Note that this is a known result for Moore-Penrose inverses. The proof used in the case of a matrix $H$ that is full rank then does not apply in the case under consideration. To prove proposition 1, we use instead a more general relationship between the derivatives of $H$ and of its Moore-Penrose inverse originally derived by Golub and Pereyra[98]

$$\frac{dH^\dagger}{dk_{ij}} = -H^\dagger \frac{dH}{dk_i j} H^\dagger + H^{\dagger 2} \frac{dH}{dk_i j}(I - HH^\dagger) + (I - HH^\dagger)\frac{dH}{dk_i j} H^{\dagger 2} \tag{50}$$

This formula is adapted from Golub and Pereyra[98] in the specific case of $H$ and $H^\dagger$ real, symmetric. In this equation, the first term on the right is the term we want. There are two additional terms, which we note as $B$ and $C$, with

$$\begin{aligned} B &= H^{\dagger 2} \frac{dH}{dk_i j}(I - HH^\dagger) \\ C &= (I - HH^\dagger)\frac{dH}{dk_i j} H^{\dagger 2} \end{aligned}$$

We need to prove that $B = C = 0$. As $C = B^T$, it is enough to prove that $B = 0$.

Let us first notice that

$$\begin{aligned} H^{\dagger 2} &= \sum_{k=N_c+1}^{3N} \sum_{l=N_c+1}^{3N} \frac{1}{\lambda_k \lambda_l} (\mathbf{e}_k \otimes \mathbf{e}_k)(\mathbf{e}_l \otimes \mathbf{e}_l) \\ &= \sum_{k=N_c+1}^{3N} \frac{1}{\lambda_k^2} \mathbf{e}_k \otimes \mathbf{e}_k \end{aligned} \tag{51}$$

as the eigenvectors $\mathbf{e}$ are orthonormal. Replacing equations 41, 49 and 51 into the definition of $B$, we get,

$$\begin{aligned} B &= \sum_{k=N_c+1}^{3N} \sum_{l=1}^{N_c} \frac{1}{\lambda_k^2} (\mathbf{e}_k \otimes \mathbf{e}_k)(\mathbf{U}_{ij} \otimes \mathbf{U}_{ij})(\mathbf{e}_l \otimes \mathbf{e}_l) \\ &= \sum_{k=N_c+1}^{3N} \sum_{l=1}^{N_c} \frac{1}{\lambda_k^2} (\mathbf{e}_k, \mathbf{U}_{ij})(\mathbf{e}_l, \mathbf{U}_{ij}) \mathbf{e}_k \otimes \mathbf{e}_l \\ &= \left( \sum_{k=N_c+1}^{3N} \frac{(\mathbf{e}_k, \mathbf{U}_{ij})}{\lambda_k^2} \mathbf{e}_k \right) \otimes \left( \sum_{l=1}^{N_c} (\mathbf{e}_l, \mathbf{U}_{ij}) \mathbf{e}_l \right) \end{aligned} \tag{52}$$

Let $\mathbf{e}_l$ be an eigenvector in the null space of $H$. Then,

$$H\mathbf{e}_l = 0 \tag{53}$$

Using equation 40, we get

$$\sum_{(i,j)} k_{ij}(\mathbf{U}_{ij} \otimes \mathbf{U}_{ij})\mathbf{e}_l = 0 \tag{54}$$

or

$$\sum_{(i,j)} k_{ij}(\mathbf{e}_l, \mathbf{U}_{ij})\mathbf{U}_{ij} = 0 \tag{55}$$

Taking the inner product with $\mathbf{e}_l$, we get

$$\sum_{(i,j)} k_{ij}(\mathbf{e}_l, \mathbf{U}_{ij})^2 = 0 \tag{56}$$

As we have assumed that all the $k_{ij}$ are strictly positive, the inner products $(\mathbf{e}_l, \mathbf{U}_{ij})$ have to be zero, for all pairs $(ij) \in \mathcal{V}$ (i.e. the set of edges in the ENM), and for all $l \in \{1, \ldots, N_c\}$. Replacing in the right most term in equation 52, we find that $B = 0$, which concludes the proof.

□

# References


1. T. Creighton, *Proteins: structures and molecular properties* (W.H. Freeman, New York, NY, 1993).

2. Z. Sun, Q. Liu, G. Qu, Y. Feng, and M. Reetz, Chem. Rev. **119**, 1626 (2019).

3. P. Sapienza and A. Lee, Curr. Opin. Pharmacol. **10**, 723 (2010).

4. B. Halle, Proc. Natl. Acad. Sci. (USA) **99**, 1274 (2002).

5. F. Zhang and R. Bruüschweiler, J. Am. Chem. Soc. **124**, 12654 (2002).

6. C.-H. Shih, S.-W. Huang, S.-C. Yen, Y.-L. Lai, S.-H. Yu, and J.-K. Hwang, Proteins: Struct. Func. Bioinfo. **68**, 34 (2007).

7. C. Lin, S. Huang, Y. Lai, S. Yen, C. Shih, C. Lu, C. Wang, and C. Hwang, Proteins: Struct. Func. Bioinfo. **15**, 929 (2008).

8. D.-W. Li and R. Bruüschweiler, Biophys. J. **96**, 3074 (2009).

9. H. Zhang, T. Zhang, K. Chen, S. Shen, J. Ruan, and L. Kurgan, Proteins: Struct. Func. Bioinfo. **76**, 617 (2009).

10. A. Schlessinger and B. Rost, Proteins: Struct. Func. Bioinfo. **61**, 115 (2005).

11. Z. Yuan, T. Bailey, and R. Teasdale, Proteins: Struct. Func. Bioinfo. **58**, 905 (2005).

12. J. Kuriyan and W. Weis, Proc. Natl. Acad. Sci. (USA) **88**, 2773 (1991).

13. P. Hünenberger, A. Mark, and W. van Gunsteren, J. Mol. Biol. **252**, 492 (1995).

14. Y. Gu, D.-W. Li, and R. Brüschweiler, J. Chem. Theory Comput. **10**, 2599 (2014).

15. Y.-P. Pang, Heliyon **2**, e00161 (2016).

16. R. Brüschweiler, J. Am. Chem. Soc. **114**, 5341 (1992).

17. A. Atilgan, S. Durell, R. Jernigan, M. Demirel, O. Keskin, and I. Bahar, Biophys. J. **80**, 505 (2001).

18. B. Erman, Biophys. J. **91**, 3589 (2006).

19. L. Yang, G. Song, and R. Jernigan, Proteins: Struct. Func. Bioinfo. **76**, 164 (2009).

20. P.-C. Chen, M. Hologne, O. Walker, and J. Hennig, J. Chem. Theory Comput. **14**, 1009 (2018).

21. G. Lipari and A. Szabo, J. Am. Chem. Soc. **104**, 4546 (1982).

22. G. Lipari and A. Szabo, J. Am. Chem. Soc. **104**, 4559 (1982).

23. S. Mahajan and Y. Sanejouand, Arch. Biochem. Biophys. **567**, 59 (2015).

24. M. Saunders and G. Voth, Annu. Rev. Biophysics **42**, 73 (2013).

25. S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. Dawid, and A. Kolinski, Chem. Rev. **116**, 7898 (2016).

26. T. Noguti and N. Go, Nature **296**, 776 (1982).

27. B. Brooks, R. Bruccoleri, and B. Olafson, J. Comp. Chem. **4**, 187 (1983).

28. M. Levitt, C. Sander, and P. Stern, J. Mol. Biol. **181**, 423 (1985).

29. B. Books, D. Janezic, and M. Karplus, J. Comp. Chem. **16**, 1522 (1995).

30. D. Case, Curr. Opin. Struct. Biol. **4**, 285 (2004).

31. M. Tirion, Phys. Rev. Lett. **77**, 1905 (1996).

32. C. Xu, D. Tobi, and I. Bahar, J. Mol. Biol. **333**, 153 (2003).

33. M. Delarue and Y.-H. Sanejouand, J. Mol. Biol. **320**, 1011 (2002).

34. Y. Wang, A. Rader, I. Bahar, and R. Jernigan, J. Struct. Biol. **147**, 302 (2004).

35. E. Dykeman and O. Sankey, J. Phys. Condens. Matter **21**, 035116 (2009).

36. E. Dykeman and O. Sankey, Phys. Rev. E **81**, 021918 (2010).

37. Y.-C. Hsieh, F. Poitevin, M. Delarue, and P. Koehl, Frontiers Bio. Sci. **3**, 85 (2016).

38. P. Koehl and M. Delarue, Prog. Biophys. Mol. Biol. **143**, 20 (2019).

39. T. Lezon, I. Shrivastava, Z. Yan, and I. Bahar, in *Handbook on Biological Networks*, edited by S. Boccaletti, V. Latora, and Y. Moreno (World Scientific Publishing Co, Singapore, 2010), pp. 129–158.

40. Y. Sanejouand, Methods Mol. Biol. **914**, 601 (2013).

41. H. Wako and S. Endo, Biophys. J. **9**, 877 (2017).

42. Y. Togashi and H. Flechsig, Int. J. Mol. Sci. **19**, 3899 (2018).

43. M. Delarue and P. Dumas, Proc. Natl. Acad. Sci. (USA) **101**, 6957 (2004).

44. E. Lindahl, C. Azuara, P. Koehl, and M. Delarue, Nucl. Acids. Res. **34**, W52 (2006).

45. F. Tama, O. Miyashita, and C. Brooks III, J. Struct. Biol. **147**, 315 (2004).

46. J. López-Blanco and P. Chacón, J. Struct. Biol. **184**, 261 (2013).

47. M. Tekpinar, Molec. Simul. **44**, 688 (2018).

48. I. Bahar, A. Atilgan, and B. Erman, Folding and Design **2**, 173 (1997).

49. T. Haliloglu, I. Bahar, and B. Erman, Phys. Rev. Lett. **79**, 3090 (1997).

50. F. Tama and Y.-H. Sanejouand, Protein Eng. **14**, 1 (2001).

51. E. Eyal, L. Yang, and I. Bahar, Bioinformatics **22**, 2619 (2006).

52. K. Hinsen, Proteins: Struct. Func. Genet. **33**, 417 (1998).

53. J. Kovacs, P. Chacon, and R. Abagyan, Proteins: Struct. Func. Bioinfo. **54**, 661 (2004).

54. L. Yang, G. Song, and R. Jernigan, Proc. Natl. Acad. Sci. (USA) **106**, 12347 (2009).

55. F. Xia, D. Tong, L. Yang, D. Wang, S. Doi, P. Koehl, and L. Lu, J. Comp. Chem. **35**, 1111 (2014).

56. F. Xia, D. Tong, and L. Lu, J. Chem. Theory Comput. **13**, 3704 (2013).

57. E. Fuglebakk, N. Reuter, and K. Hinsen, J. Chem. Theory Comput. **9**, 5618 (2013).

58. K. Hinsen, Bioinformatics **24**, 521 (2008).

59. D. Riccardi, Q. Cui, and G. Phillips Jr, Biophys. J. **96**, 464 (2009).

60. R. Soheilifard, D. Makarov, and G. Rodin, Phys. Biol. **5**, 026008 (2008).

61. J. Hafner and W. Zheng, J. Chem. Phys. **132**, 014111 (2010).

62. T. Lezon, Proteins: Struct. Func. Bioinfo. **80**, 1133 (2012).

63. D. Jacobs, A. Rader, L. Kuhn, and M. Thorpe, Proteins: Struct. Func. Genet. **44**, 150 (2001).

64. A. Rader, B. Hespenheide, L. Kuhn, and M. Thorpe, Proc. Natl. Acad. Sci. (USA) **99**, 3540 (2002).

65. H. Edelsbrunner and E. P. Mücke, ACM Trans. Graphics **13**, 43 (1994).

66. J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam, Proteins: Struct. Func. Genet. **33**, 1 (1998).

67. H. Edelsbrunner and P. Koehl., Discrete and Computational Geometry (MSRI Publications) **52**, 243 (2005).

68. P. Koehl, J. Chem. Theory Comput. **14**, 3903 (2018).

69. Y. Dehouck and U. Bastolla, Integrative biology **9**, 627 (2017).

70. V. Schomaker and K. Trueblood, Acta Cryst. B **24**, 63 (1969).

71. R. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM J. Sci. Comput. **16**, 1190 (1995).

72. T. Meyer, M. D'Abramo, A. Hospital, M. Rueda, C. Ferrer-Costa, A. Pérez, O. Carrillo, J. Camps, C. Fenollosa, D. Repchevsky, et al., Structure **18**, 1399 (2010).

73. D. Case, T. Cheatham III, T. Darden, H. Gohlke, R. Luo, K. Merz Jr., A. Onufriev, C. Simmerling, B. Wang, and R. Woods, J. Comp. Chem. **26**, 1668 (2005).

74. W. Jorgensen, J. Chandrasekhar, J. Madura, R. Impey, and M. Klein, J. Chem. Phys. **79**, 926 (1983).

75. A. Bhattacharyya, Bull. Calcutta Math. Soc. **35**, 99 (1943).

76. A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, in *2011 International Conference on Computer Vision (ICCV)* (2011), pp. 2399–2406.

77. E. Fuglebakk, J. Echave, and N. Reuter, Bioinformatics **28**, 2431 (2012).

78. A. Zomorodian, L. Guibas, and P. Koehl, Comput. Aided Geom. Design **23**, 531 (2006).

79. M. Levitt, Angew. Chem. Int. Ed. Engl. **53**, 10006 (2014).

80. C. Micheletti, P. Carloni, and A. Maritan, Proteins: Struct. Func. Bioinfo. **55**, 635 (2004).

81. M. Rueda, P. Chacon, and M. Orozco, Structure **15**, 565 (2007).

82. L. Orellana, M. Rueda, C. Ferrer-Costa, J. Lopez-Blanco, P. Chacon, and M. Orozco, J. Chem. Theory Comput. **6**, 2910 (2010).

83. N. Leioatts, T. Romo, and A. Grossfield, J. Chem. Theory Comput. **8**, 2424 (2012).

84. D. Tobi and I. Bahar, Proc. Natl. Acad. Sci. (USA) **102**, 18908 (2005).

85. R. Mendez and U. Bastolla, Phys. Rev. Lett. **104**, 228103 (2010).

86. U. Bastolla and Y. Dehouck, J. Chem. Inf. Model. **59**, 4929 (2019).

87. S. Le Grand and K. Merz, J. Comp. Chem. **14**, 349 (1993).

88. M. Vihinen, E. Torkkila, and P. Riikonen, Proteins: Struct. Func. Genet. **19**, 141 (1994).

89. P. Karplus and G. Schulz, Naturwissenschaften **72**, 212 (1985).

90. D. Smith, P. Radivojac, Z. Obradovic, A. Dunker, and G. Zhu, Protein Sci. **12**, 1060 (2003).

91. P. Romero, Z. Obradovic, C. Kissinger, J. Villafranca, and A. Dunker, in *Int. Conf. Neural Net.* (1997), pp. 90–95.

92. P. Koehl, F. Poitevin, R. Navaza, and M. Delarue, J. Chem. Theory Comput. **13**, 1424 (2017).

93. D. Jacobs, L. Kuhn, and M. Thorpe, in *Rigidity theory and applications*, edited by M. Thorpe and P. Duxbury (Kluwer Academic, New York, 1999), pp. 357–384.

94. O. Carugo and P. Argos, Protein Sci. **6**, 2261 (1997).

95. J. Helliwell, Crystallogr. Rev. **14**, 189 (2008).

96. R. Diamond, Acta Crystallogr. A **46**, 425 (1990).

97. A. Kidera and N. Go, Proc. Natl. Acad. Sci. (USA) **87**, 3718 (1990).

98. G. Golub and V. Pereyra, SIAM J. Num. Anal. **10**, 413 (1973).