Damped-Dynamics Flexible Fitting

Julio A. Kovacs,* Mark Yeager,[†‡§] and Ruben Abagyan*
*Department of Molecular Biology, The Scripps Research Institute, La Jolla, California; [†]Department of Molecular Physiology and Biological Physics, University of Virginia Health System, Charlottesville, Virginia; [‡]Department of Cell Biology, The Scripps Research Institute, La Jolla, California; and [§]Division of Cardiovascular Diseases, Scripps Clinic, La Jolla, California

ABSTRACT In fitting atomic structures into EM maps, it often happens that the map corresponds to a different conformation of the structure. We have developed a new methodology to handle these situations that preserves the covalent geometry of the structure and allows the modeling of large deformations. The first goal is achieved by working in generalized coordinates (positional and internal coordinates), and the second by avoiding harmonic potentials. Instead, we use dampers (shock absorbers) between every pair of atoms, combined with a force field that attracts the atomic structure toward incompletely occupied regions of the EM map. The trajectory obtained by integrating the resulting equations of motion converges to a conformation that, in our validation cases, was very close to the target atomic structure. Compared to current methods, our approach is more efficient and robust against wrong solutions and to overfitting, and does not require user intervention or subjective decisions. Applications to the computation of transition pathways between known conformers, homology and loop modeling, as well as protein docking, are also discussed.

INTRODUCTION

One of the most widespread approaches to model atomic structures at high resolution based on EM maps has been the rigid-body fitting of known atomic structures of the component chains into the EM map. However, when there are significant conformational differences between the known structures and the map, this classical rigid-body approach yields unsatisfactory results. This has triggered an intense research on methodologies that can account for the flexibility of the molecules involved.

Various lines have been explored thus far to tackle this challenging problem. One of them, developed by Wriggers and co-workers, is based on the concept of vector quantization, whereby reduced models of both the atomic structure and the EM map are computed, and then a molecular-dynamics (MD) simulation is performed with an extra energy term that penalizes the deviations between the corresponding codebook vectors, bringing them into register (1). This method has been implemented in the Situs package (2), and refined by the addition of ''skeletons'' to suppress inessential degrees of freedom (DOF), improving its robustness against experimental noise (3). This approach has been successfully applied to important cases, such as RNA polymerase (4) and actin (5).

The disadvantages of this approach are:

1. It is computationally costly to perform the MD simulation.
2. Distortions of the stereochemistry are likely, due to the strong pull between codebook vectors combined with a Cartesian MD.
3. Local rearrangements are usually not captured due to the coarse-grain representations.
4. User intervention is required to define two aspects of the simulation:
   —Number of codebook vectors to be used. An adequate trade-off number has to be chosen, because too few of them will not give the structure a sufficient number of degrees of freedom, and too many of them would produce overfitting and noise-sensitivity.
   —Use of skeletons. When using skeletons, user's inspection and judgment are necessary to decide which pairs of codebook vectors should be distance-restrained.

Another line to approach flexibility is by utilizing normal-mode analysis (NMA). Preliminary explorations in this regard, in which both the atomic structure and the EM map are vector-quantized and subjected to NMA, have been reported (5–7). These works laid the ground for the development of specific algorithms that use the normal modes of the atomic structure to deform it so as to maximize the cross-correlation with the EM map (8–10). More recently these algorithms have been implemented in a fitting tool called *NORMA* (11), which performs the cross-correlation calculations in reciprocal space (as in (10)). A related but different technique has been proposed by Hinsen et al. (12): they define a force field (as the gradient of the misfit energy function) that pulls the atomic structure toward regions of high density of the map, using the normal modes to compute, in a convenient way, the displacements produced by the forces.

The disadvantages of the NMA approach are:

1. The normal modes are computed in Cartesian coordinates, which necessarily causes distortions to the covalent geometry of the structure.

Submitted February 25, 2008, and accepted for publication June 5, 2008.

Address reprint requests to Julio Kovacs, Tel.: 858-746-1135; E-mail: jkovacs@seaspace.com; or to Ruben Abagyan, Tel.: 858-784-8595; E-mail: abagyan@scripps.edu.

Julio A. Kovacs' present address is SeaSpace Corporation, 12120 Kear Pl., Poway, CA 92064.

Editor: Ruth Nussinov.

0006-3495/08/10/3192/16 $2.00

doi: 10.1529/biophysj.108.132357

2. User decision (or a heuristic argument) is needed to choose the optimal number of modes to be used (as before, too many would produce overfitting).
3. There is a significant chance of getting trapped in a local maximum of the cross-correlation function.

A method quite similar to that of Hinsen et al. (12) had been proposed earlier by Chen et al. (13). Originally developed for x-ray crystallography refinement, it consists in minimizing an energy function that includes the misfit between the structure and the map and energy terms related to the stereochemical properties of the model. (A harmonic potential is used in (12) in place of the latter terms.) The main difficulty in applying this method lies in avoiding local minima of the energy function. Also, to reduce the number of DOF, the authors treat the domains of the molecule as rigid bodies. This entails user intervention to fragment the molecule. (See also (14) for an improvement of this method and a review of a number of approaches to rigid-body and flexible fitting.)

A rather different approach combines comparative protein structure modeling (homology modeling) with EM data (reviewed in (15)). For cases when experimentally determined atomic structures of components are not available (or their conformation is significantly different from the one present in the EM map), Topf et al. (16) proposed a method consisting in the determination of homology models of each component by considering a number of different alignments between the target sequence and a related template structure, and subsequently assessing the resulting structures by how well they fit (rigidly) into the EM map. A variation of this approach uses families of known conformations of the protein domains to perform a principal-component analysis, from which deformed models are generated, without using the given EM map (17). These models are subsequently ranked according to their cross-correlation with the EM map. The main downside of this method is its dependence upon the availability of a sufficient number of dissimilar structures in the superfamilies. It also requires user intervention and a careful procedure to restore the correct stereochemistry after obtaining the deformed models.

Very recently, two articles came out which use a Monte Carlo approach to drive the structure toward the EM map so as to increase the cross-correlation function. The first one, by Topf et al. (18), uses, as variables to be optimized, the position and orientation of various fragments in which the structure is divided. These fragments are made smaller as the simulation progresses, but to make the procedure automated, the authors used the domains of the original structure and then the secondary-structure elements (implying that the latter will never bend). These rigid fragments are independently moved by the Monte Carlo procedure, after which they, as well as the linkers, are refined by conjugate-gradient minimization followed by simulated annealing with molecular dynamics.

The second one, by Jolley et al. (19), is based on FRODA (20), a Monte Carlo-type algorithm which at each step throws the atoms from their current positions by a certain amount, and then refits them using geometric constraints (using the FIRST algorithm (21)) to obtain a new valid conformation (i.e., one that again satisfies the constraints). This new conformation is then cross-correlated with the given EM map, and accepted or rejected using a standard Metropolis criterion. Similarly to the above method, the authors chose the parameters of the noncovalent constraint network so as to keep the secondary-structure elements rigid, to prevent loosing their geometric integrity.

We should point out that the idea of using Monte Carlo and local minimization to optimize a combination of energy terms (including cross correlation with a map) is not new, although the above two articles seem to be the first ones to show applications in EM fitting. This approach has been implemented in the ICM software package (22), which utilizes internal variables to describe the mechanics and dynamics of biomolecular complexes (23).

In an attempt to address the weaknesses of existing flexible-fitting methods, we have developed an approach called damped-dynamics flexible fitting (DDFF). A dynamical system is defined by placing dampers (shock absorbers) between every pair of atoms within a cutoff distance (similarly to what is done in NMA with springs). A force field acting on this system is defined in such a way as to attract the atoms toward nonfully occupied (density-wise) regions of the EM map. The resulting equations of motion are integrated in generalized coordinates. These coordinates consist of six positional coordinates for each chain, plus internal coordinates given by the torsion angles ϕ, ψ, and χ. Bond lengths, bond angles, and peptide ω-angles are kept fixed. This ensures that the covalent geometry is preserved throughout. The stereochemical structure is maintained by using distance-dependent damping coefficients (larger for close-by atoms). The system is made completely damped by setting the atom masses to 0 and by adding a drag term to the equations. As a consequence, the resulting equations are of first order and linear in the derivatives, making their numerical integration simple and efficient, and avoiding oscillations and transients of the trajectories. The efficiency was dramatically increased upon the implementation of a recursive method to compute the system matrix (24), which reduces the complexity from $O(N^4)$ to $O(N^2)$, where N is the number of atoms. Also, the storage requirements of the code were streamlined to be linear in N (except for the system matrix).

Our approach does not require user intervention or adjustment of parameters on a case-by-case basis. The few parameters that do appear in the method are fixed and hidden from the user. We use a reduced residue model by coalescing all atoms of each side chain beyond the C_β into one pseudoatom (Fig. 1 *a*). In this way, at most three torsion angles are needed for each residue. Other than this, no reductions are made to the number of DOF in the system, allowing local conformational changes to be captured. To compensate for the inaccuracy introduced by the reduced residue model, a

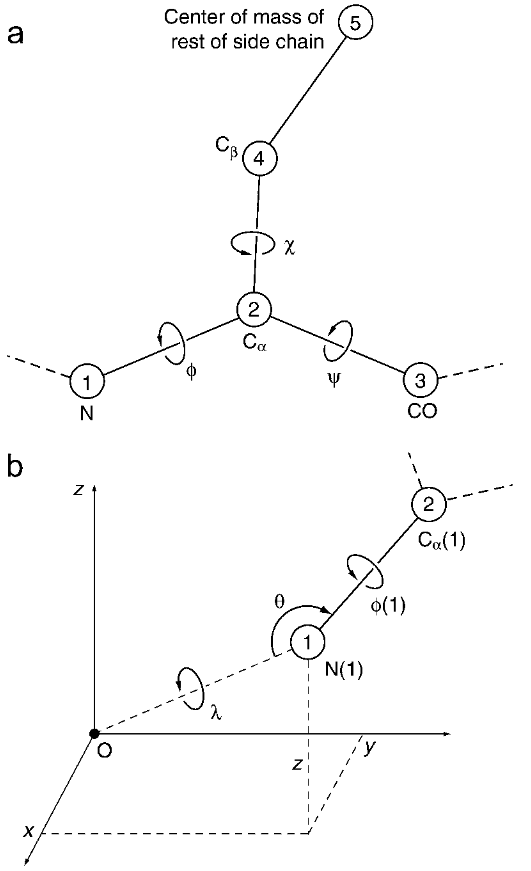

FIGURE 1 (*a*) Simplified residue model showing the internal variables ϕ, ψ, and χ. Pseudo-atoms 1, 2, 4, and 5 include the corresponding hydrogen atoms. For Ala, pseudo-atom 5 is absent; for Gly, both pseudo-atoms 4 and 5 are absent. In these two cases, there is no χ-angle. Also, there is no ϕ for the first residue of the chain or for prolines, and there is no ψ for the last residue of the chain. (*b*) Positional variables of each chain: Cartesian coordinates (x, y, z) of the first atom of the chain, spherical coordinates (λ, θ) of the second atom of the chain relative to the first, and dihedral angle $\phi(1)$. The number *1* means that this is the first residue of the chain, to which these positional variables are ascribed.

global side-chain prediction is performed periodically along the trajectory (25). This also prevents the fitting process from getting stuck or astray due to wrong side-chain conformations that would inevitably occur if they were simply evolved from their initial conformations.

Finally, as we argue in Discussion and Conclusions, our method should be more immune against getting stuck in ‘‘local minima’’ than other approaches, and provides an effective way to control overfitting.

METHODS

General equations of motion

Let our atomic structure consist of N atoms, which may be distributed among several chains. We start by writing down Newton's equation for an atom k,

$$m_k \ddot{\mathbf{r}}_k = \mathbf{F}_k^{(a)} + \mathbf{F}_k^{(c)}, \tag{1}$$

where $\mathbf{r}_k$ = position of atom k, m_k = mass of atom k, $\mathbf{F}_k^{(a)}$ = applied force, and $\mathbf{F}_k^{(c)}$ = constraint force.

In our problem, the constraints represent the assumption of fixed bond lengths, bond angles, and ω-peptide angles. Furthermore, we use a simplified residue model whereby each residue is represented by three backbone pseudo-atoms and two side-chain pseudo-atoms (Fig. 1 *a*). Thus, three dihedral angles are needed for each residue: ϕ, ψ, χ (with the exceptions of Ala, Gly, and Pro, and the last residue of each chain). Also, five additional variables (x, y, z, λ, θ) define, along with the ϕ of the first residue of each chain, the position and orientation of the chain (Fig. 1 *b*). Finally, some of these coordinates may, according to the specific application, be kept fixed. We denote the set of all the free variables by $q_1,\ldots,q_M$, M being the total number of them.

The derivation of the following equations is general and independent of the particular residue model used.

To convert the above fundamental equations (Eq. 1) to generalized coordinates, we scalarly multiply Newton's equations by the derivatives $\partial \mathbf{r}_k / \partial q_j$ (which form the so-called Wilson's matrix) and add them up over k:

$$\sum_k m_k \left\langle \ddot{\mathbf{r}}_k, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle = \sum_k \left\langle \mathbf{F}_k^{(a)}, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle + \sum_k \left\langle \mathbf{F}_k^{(c)}, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle.$$

(The angle brackets denote scalar (inner or ‘‘dot’’) product.) The left-hand side can be written as (26)

$$\sum_k m_k \left\langle \ddot{\mathbf{r}}_k, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j},$$

where

$$T = \frac{1}{2}\sum_k m_k \|\dot{\mathbf{r}}_k\|^2$$

is the kinetic energy of the system. Therefore, the equations of motion become

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} = Q_j + G_j, \tag{2}$$

where

$$Q_j = \sum_k \left\langle \mathbf{F}_k^{(a)}, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle, \quad G_j = \sum_k \left\langle \mathbf{F}_k^{(c)}, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle.$$

We will assume that the forces of constraint, $\mathbf{F}_k^{(c)}$, do no virtual work. Even though we do not have a rigorous proof of this fact, heuristically this seems to be the case in our problem—and is confirmed by the results. This implies

$$\sum_j G_j \cdot \delta q_j = 0, \tag{3}$$

where the δq_j values are the virtual displacements (26). If, in addition, the coordinates q_j are independent, Eqs. 2 and 3 imply the standard equations of motion:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} = Q_j \quad (j = 1,\ldots,\mathrm{M}). \tag{4}$$

However, we need to consider situations in which the generalized coordinates q_j are not independent. An important example is when we want to keep the endpoints of a loop, or of any given stretch of the molecule, fixed in space. These situations are represented by relations among the q_j values of the form

$$f_\alpha(q_1,\ldots,q_M) = 0 \quad (\alpha = 1,\ldots,K). \tag{5}$$

In this case, the expressions in Eq. 5 are no longer valid, and we need to use a more general variational principle to derive the corresponding equations of motion. One such principle—a generalization of Hamilton's principle to nonconservative systems—is (27)

$$\int_{t_1}^{t_2} \left[\delta T + \sum_j (Q_j + G_j)\delta q_j \right] dt = 0.$$

Here we introduce the relations by means of Lagrange multipliers (26):

$$\int_{t_1}^{t_2}\left[\delta\left(T+\sum_{\alpha}h_{\alpha}f_{\alpha}\right)+\sum_{j}(Q_j+G_j)\delta q_j\right]dt=0. \tag{6}$$

(The Lagrange multipliers h_α are, like the q_j, functions of t.)

The variation δ is over all paths that keep the endpoints fixed, and is to be taken over all variables: q_j and h_α. By doing so, and taking Eq. 3 into account, Eq. 6 becomes

$$\int_{t_1}^{t_2}\left\{\sum_{j}\left[\frac{\partial T}{\partial q_j}-\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right)+\sum_{\alpha}h_{\alpha}\frac{\partial f_{\alpha}}{\partial q_j}+Q_j\right]\cdot\delta q_j + \sum_{\alpha}f_{\alpha}\delta h_{\alpha}\right\}dt=0.$$

Equating each term to 0 (which is now possible because of the presence of the terms involving δh_α), we get the general equations of motion:

$$\left.\begin{array}{rl} \dfrac{d}{dt}\left(\dfrac{\partial T}{\partial \dot{q}_j}\right)-\dfrac{\partial T}{\partial q_j} = Q_j+\displaystyle\sum_{\alpha}h_{\alpha}\frac{\partial f_{\alpha}}{\partial q_j} & \forall j \\ f_{\alpha}=0 & \forall \alpha \end{array}\right\}. \tag{7}$$

Damped-dynamics equations of motion

Now we want to specialize to the case in which

$$\mathbf{F}^{(a)}=\mathbf{F}^{(s)}+\mathbf{F}^{(d)}+\mathbf{F}^{(m)},$$

where $\mathbf{F}^{(s)}$ = damper force, $\mathbf{F}^{(d)}$ = drag force, and $\mathbf{F}^{(m)}$ = density-map force, and in which all $m_k = 0$. This implies that the atoms will always move at their limiting speeds, and prevents transient phenomena from occurring. These various forces are calculated in the following paragraphs.

Damper force

The force produced by the dampers is

$$\mathbf{F}_k^{(s)}=-\sum_{\substack{l=1\\ l\neq k}}^{N}C_{kl}\frac{\langle\dot{\mathbf{r}}_k-\dot{\mathbf{r}}_l,\mathbf{r}_k-\mathbf{r}_l\rangle}{\|\mathbf{r}_k-\mathbf{r}_l\|^2}(\mathbf{r}_k-\mathbf{r}_l),$$

where the C_{kl} values are the (distance-dependent) damping coefficients. We chose the dependence

$$C_{kl}=\begin{cases}\dfrac{C_{kl}^{0}}{\|\mathbf{r}_k-\mathbf{r}_l\|^{1/2}} & \text{if} \quad \|\mathbf{r}_k-\mathbf{r}_l\|<d_{cut} \\ 0 & \text{otherwise,}\end{cases}, \tag{8}$$

where d_{cut} is a cutoff distance, initially set to half of the diameter of the atomic structure. During the trajectory this cutoff distance is periodically reduced, as described in Numerical Integration Scheme, below.

A simple calculation shows that the corresponding generalized forces are

$$Q_j^{(s)}=\sum_{k}\left\langle\mathbf{F}_k^{(s)},\frac{\partial\mathbf{r}_k}{\partial q_j}\right\rangle=-\sum_{i}V_{ij}\dot{q}_i,$$

where

$$V_{ij}=\sum_{k<l}\frac{C_{kl}}{\|\mathbf{r}_k-\mathbf{r}_l\|^2}\cdot\left\langle\frac{\partial(\mathbf{r}_k-\mathbf{r}_l)}{\partial q_i},\mathbf{r}_k-\mathbf{r}_l\right\rangle \times\left\langle\frac{\partial(\mathbf{r}_k-\mathbf{r}_l)}{\partial q_j},\mathbf{r}_k-\mathbf{r}_l\right\rangle. \tag{9}$$

Drag force

The force produced by water resistance is $\mathbf{F}_k^{(d)}=-b_k\dot{\mathbf{r}}_k$, where the b_k values are the friction constants. The corresponding generalized forces are

$$Q_j^{(d)}=\sum_{k}\left\langle\mathbf{F}_k^{(d)},\frac{\partial\mathbf{r}_k}{\partial q_j}\right\rangle=-\sum_{i}B_{ij}\dot{q}_i,$$

where

$$B_{ij}=\sum_{k}b_k\left\langle\frac{\partial\mathbf{r}_k}{\partial q_i},\frac{\partial\mathbf{r}_k}{\partial q_j}\right\rangle. \tag{10}$$

Density-map force

This force, $\mathbf{F}^{(m)}$, will be the resultant from the forces produced by the unoccupied regions of the density map. (See Force Field, below.) The corresponding generalized forces are

$$Q_j^{(m)}=\sum_{k}\left\langle\mathbf{F}_k^{(m)},\frac{\partial\mathbf{r}_k}{\partial q_j}\right\rangle.$$

Substituting all of the above into the general equations of motion (Eq. 7), we obtain the DD equations of motion

$$\left.\begin{array}{rl} \displaystyle\sum_{i}(B_{ij}+V_{ij})\dot{q}_i = Q_j^{(m)}+\sum_{\alpha}h_{\alpha}\frac{\partial f_{\alpha}}{\partial q_j} & \forall j \\ f_{\alpha}=0 & \forall\alpha \end{array}\right\}. \tag{11}$$

Differentiating $f_\alpha = 0$ with respect to t, we get

$$\sum_{i}\frac{\partial f_{\alpha}}{\partial q_i}\dot{q}_i=0.$$

Hence, our system (Eq. 11) (for the unknowns $\dot{q}_i$, h_α) becomes

$$\left.\begin{array}{rl} \displaystyle\sum_{i}(B_{ij}+V_{ij})\dot{q}_i-\sum_{\alpha}\frac{\partial f_{\alpha}}{\partial q_j}h_{\alpha}=Q_j^{(m)} & \forall j \\ \displaystyle\sum_{i}\frac{\partial f_{\alpha}}{\partial q_i}\dot{q}_i=0 & \forall\alpha \end{array}\right\}$$

or, in matrix form

$$\left(\begin{array}{ccccc} & & \dfrac{\partial f_1}{\partial q_1} & \cdots & \dfrac{\partial f_K}{\partial q_1} \\ & \mathrm{B}+\mathrm{V} & \vdots & & \vdots \\ & & \dfrac{\partial f_1}{\partial q_M} & \cdots & \dfrac{\partial f_K}{\partial q_M} \\ \dfrac{\partial f_1}{\partial q_1} & \cdots & \dfrac{\partial f_1}{\partial q_M} & 0 \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots \\ \dfrac{\partial f_K}{\partial q_1} & \cdots & \dfrac{\partial f_K}{\partial q_M} & 0 \cdots & 0 \end{array}\right)\cdot\left(\begin{array}{c} \dot{q}_1 \\ \vdots \\ \dot{q}_M \\ \hline -h_1 \\ \vdots \\ -h_K \end{array}\right) = \left(\begin{array}{c} Q_1^{(m)} \\ \vdots \\ Q_M^{(m)} \\ \hline 0 \\ \vdots \\ 0 \end{array}\right). \tag{12}$$

After solving this system of equations for $\dot{q}_\mathrm{i}$ and h_α, the updated values of q_i are obtained by making an Euler step,

$$\Delta q_\mathrm{i} = \dot{q}_\mathrm{i}\Delta t.$$

This gives a new conformation, on which the process is repeated iteratively, yielding the trajectory. Details on the numerical integration scheme are given later.

Computation of the matrices V and B

The direct way of computing these matrices, namely by using Eqs. 9 and 10, would make our approach exceedingly slow (of the order of N^4). We have developed recursive algorithms to compute these matrices efficiently, in $O(N^2)$ operations. These algorithms are described in Appendix A.

Force field

The right-hand side of the DD equations of motion, Eq. 12, contains the EM-map force field $\mathbf{F}^{(\mathrm{m})}$ through the transformation $Q_\mathrm{j}^{(\mathrm{m})} = \sum_\mathrm{k}\langle \mathbf{F}_\mathrm{k}^{(\mathrm{m})}, (\partial \mathbf{r}_\mathrm{k}/\partial \mathrm{q}_\mathrm{j})\rangle$. We shall now give the analytical definition of this force field $\mathbf{F}^{(\mathrm{m})}$ in terms of both the current conformation of the atomic structure and the density values of the EM map.

The definition of the $\mathbf{F}_\mathrm{k}^{(\mathrm{m})}$ follows these rules:

—Atoms are attracted by nonfully occupied regions of the map (i.e., where $f(p) - g(p) > 0$).

—Atoms are repelled by overoccupied regions of the map (i.e., where $f(p) - g(p) < 0$).

Here, f denotes the EM density map, and g the structure-induced density map, as defined next.

Structure-induced map

The first step is to lower the resolution of the atomic structure, by convolving it with a Gaussian kernel whose standard deviation σ is in accordance with the nominal resolution R of the given EM map f: $\sigma = R/(2\sqrt{3})$ (9),

$$g(\mathbf{p}) = c\sum_\mathrm{k} w_\mathrm{k}^{\mathrm{at}} e^{-\frac{\|\mathbf{p}-\mathbf{r}_\mathrm{k}\|^2}{2\sigma^2}},$$

where $w_\mathrm{k}^{\mathrm{at}}$ is the atomic weight of atom k, and c is a normalization constant so that $\int g = 1$. Also, the given EM density map f is thresholded at a user-specified level and then normalized so that $\int f = 1$. Hydrogen atoms were excluded in the computation of $g(\mathbf{p})$.

Force-field definition

The force acting on atom k is defined as

$$\mathbf{F}_\mathrm{k}^{(\mathrm{m})} = 0.1\, w_\mathrm{k}^{\mathrm{at}} \cdot \int_{\{\mathbf{p}|\mathrm{f}(\mathbf{p})>0\}} (f(\mathbf{p}) - g(\mathbf{p})) \times A(\|\mathbf{p} - \mathbf{r}_\mathrm{k}\|) \cdot (\mathbf{p} - \mathbf{r}_\mathrm{k}) d\mathbf{p}. \tag{13}$$

The first factor inside the integral implements the rules stated above. The factor $(\mathbf{p} - \mathbf{r}_\mathrm{k})$ expresses the fact that the force exerted by a point $\mathbf{p}$ on atom k is directed from the atom to the point. The function $A(r)$ is defined as

$$A(r) = \begin{cases} 1/r_0^2 & \text{for} \quad r \leq r_0 \\ 1/r^2 & \text{for} \quad r > r_0 \end{cases},$$

and is included so that $r\,A(r)$ goes as r for small values of $r = \|\mathbf{p} - \mathbf{r}_\mathrm{k}\|$, and as $1/r$ for large r. In this way, if $\mathbf{p}$ is far from $\mathbf{r}_\mathrm{k}$, this atom is attracted little by $\mathbf{p}$, and, when $\mathbf{r}_\mathrm{k}$ approaches $\mathbf{p}$, the force gets proportional to the distance. The parameter r_0 (where the maximum of $r\,A(r)$ is attained) was set to $r_0 = 1.5$ Å.

Numerical integration scheme

The computation of the force field, at each time step, is the most costly part of our algorithm. (The computation of the structure-induced map was quite alleviated by precomputing the Gaussian kernel on a grid of radius 3σ. The computation of the matrices **V** and **B**, by using the recursive algorithm, is not an issue.) Therefore it is important to implement an efficient numerical scheme to optimize the time step, adjusting it along the simulation to be the longest possible.

Time-step control

As pointed out right after Eq. 12, once the $\dot{q}_\mathrm{i}$ are obtained using that equation, the q_i are updated by making an Euler step: $\Delta q_\mathrm{i} = \dot{q}_\mathrm{i}\Delta t$. The value of Δt used for this is determined as follows.

We use the velocity vector fields $(\dot{\mathbf{r}}_1, \ldots, \dot{\mathbf{r}}_\mathrm{N})$ in the previous and the current time steps, which are denoted here by **a** and **b**, respectively. These are considered as $3N$-dimensional vectors, with root-mean square (RMS) norms a and b, and a subtended angle α between them, determined by $\cos\alpha = \langle \mathbf{a}, \mathbf{b}\rangle/ab$.

The time step is updated indirectly by first updating the RMS displacement D_T that the structure should undergo. (T = step number.) We use the following algorithm to update D_T:

$$D_0 = \text{half of the user-requested minimum displacement}$$
$$\text{between consecutive output conformations,}$$
$$D_\mathrm{T} = \frac{D_{\mathrm{T}-1}}{\frac{\mathrm{a}}{\mathrm{b}} - \cos\alpha}, \quad \text{for} \quad T > 0. \tag{14}$$

Then the time increment to be used is determined by

$$\Delta t = \frac{D_\mathrm{T}}{b}. \tag{15}$$

The rationale of the above formula is somewhat heuristic. It is obtained as the result of an extrapolation, by first projecting the previous-step velocity **a** (orthogonally to its direction) unto the straight line passing through **b** (which has its origin at a point $D_{\mathrm{T}-1}$ Å from **a**'s origin), thereby adopting a length $a/\cos\alpha$. A linear extrapolation is now performed to find the point (on **b**'s line) where the velocity should become 0.

Some safeguards are applied before using the D_T as given above. First, a cap of 1.2 is enforced on the ratio $D_\mathrm{T}/D_{\mathrm{T}-1}$ (including the case $a/b \leq \cos\alpha$). Second, D_T is decreased, if necessary, so that no atom will move >4 Å in the step.

Conformation output

The user specifies a minimum RMS displacement between output conformations, so as to avoid writing out a large number of similar sets of conformations. (For most of the examples shown in this article, we used 1 Å.) When the sum of the RMS displacements in consecutive steps reaches the specified minimum, the current conformation is written to a file, after being side-chain-optimized by means of SCATD (25). This program performs a side-chain prediction using a very fast tree-decomposition algorithm that furnishes the best side-chain conformations for a given backbone geometry, by minimizing a simplified van der Waals potential. This step is important to escape from wrong side-chain conformations that would inevitably occur if they were simply evolved from their initial conformations.

Convergence criterion

At the beginning of the simulation, a velocity threshold ε is defined as $\varepsilon = 10^{-4}/\rho$, where ρ is the diameter of the atomic structure in Å (the proper units are included in the factor 10^{-4}). This threshold is used to compare b in the convergence test below. At each time step, it is first checked whether $b = 0$. If so, the simulation ends. If $b \neq 0$, the following test is performed:

Is Mean $\{D_{\mathrm{T}-5}, \ldots, D_\mathrm{T}\} < 0.1$ Å?

- **No.** Generate a new conformation by updating the variables through an Euler step using the Δt from Eq. 15,

$$\Delta q_{\mathrm{j}} = \dot{q}_{\mathrm{j}} \Delta t,$$

and perform a new step.

- **Yes.** Is $b < \varepsilon$ and did d_{cut} reach 1/8 of its initial value?

— **No.** Divide the cutoff distance d_{cut} by 2, but without going below 1/8 of its initial value. Update variables and perform a new step, as above.

— **Yes.** End the simulation.

As safeguards, the simulation stops whenever the number of output conformations reaches 100, as well as if the number of steps performed since the previous output conformation reaches 100, since this would signal a very slow motion of the structure (still above the velocity threshold ε).

RESULTS

Validation tests

We have validated our approach by applying it to two cases with simulated EM maps: Actin and Spk1. The simulated maps were generated in the same way as described above for the structure-induced map.

For each of these cases, we took two conformations and used one of them to build a simulated map, to which the other conformation was then fitted.

Actin

Actin plays a critical role in the shape and internal structure of cells, as well as in their motions. Together with myosin it participates in muscle contraction. It can exist in globular form (actin monomer, or G-actin) or in filament form (actin polymer, or F-actin). These filaments twine around each other, forming a cytoskeleton, which provides a scaffold for the cell's organization (28).

For this validation test, we used two model structures of the actin monomer taken from the Situs website (2,29), which we call here the ''open'' and the ''closed'' conformation. For this particular case, we made fittings in both directions: from open to closed and from closed to open, to assess the ability of our method to open closed structures. In each direction, the target structure was used to generate a simulated density map, at 15 Å resolution, into which the starting conformation was fitted (Fig. 2). These actin structures have 375 residues, and the total number of free variables was 1034. The simulation took ~39 min on a 1.1-GHz AMD Opteron Linux PC.

Fig. 2, *e* and *f*, show the evolution of the overlap between each conformation-induced map and the target map, and the RMS deviation (RMSD) between each conformation and the target conformation (used only to generate the simulated

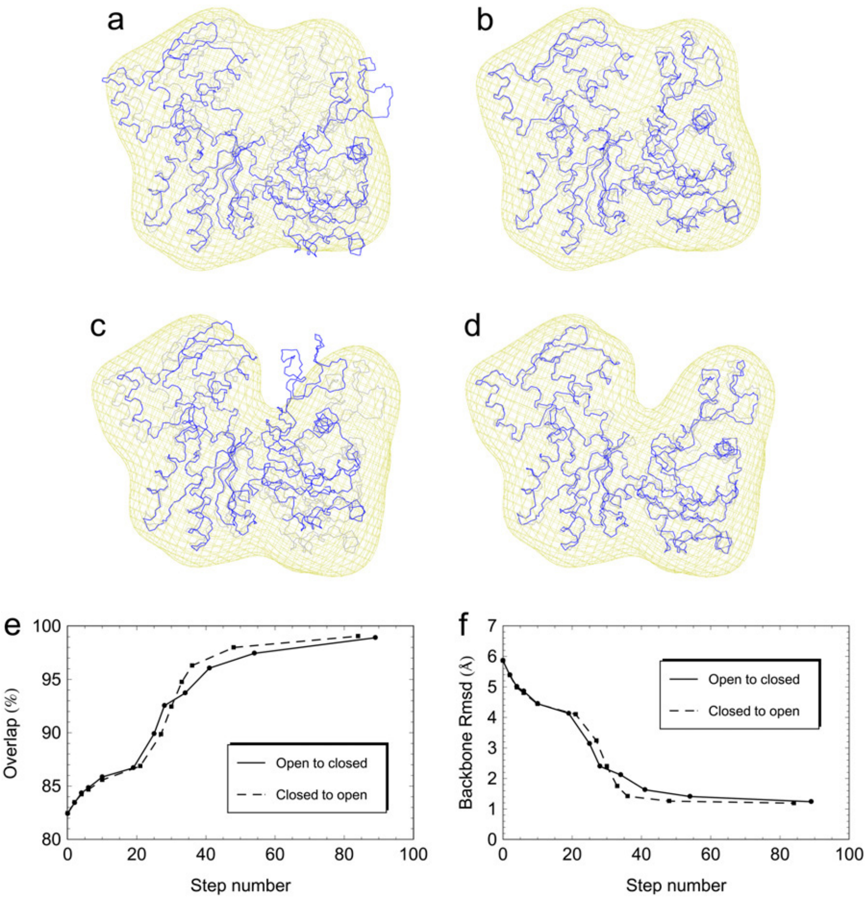

FIGURE 2 Flexible fitting of each of the two conformations (open and closed) of the actin molecule into a simulated density map (*yellow wireframe*, 1σ above the mean) at 15 Å resolution generated by the other one (shown in *light gray*). (*a*) The open conformation of actin is shown in blue. (*b*) The final conformation of the trajectory toward the closed conformation is displayed in blue. (*c*) The closed conformation is shown in blue. (*d*) The final conformation of the trajectory toward the open conformation is displayed in blue. (*e*) Evolution of the overlap values along the trajectories. (*f*) Evolution of the backbone RMSD values along the trajectories.

map), for each of the two directions of the fitting. The overlap κ between the density maps f and g is defined by

$$\kappa = 1 - \frac{\int |f - g|}{\int f + \int g}.$$

The plots show that the overlap, for each of the directions, goes from ~82% to ~99%, while the RMSD goes from ~6 Å down to just over 1 Å. These RMSD values include rigid motions, not only net deformations.

Spk1

Spk1 (for serine-protein kinase) is an enzyme from *Saccharomyces cerevisiae* that phosphorylates proteins on serine, threonine and, to a lesser degree, on tyrosine (30).

For this validation test, we used an NMR structure with its ''arms'' in a ''closed'' conformation as the target (PDB accession code 1K3Q), from which a simulated density map was generated at 15 Å resolution, and a minimized average structure (with ''arms open,'' PDB accession code 1J4P) as the starting conformation (30) to be fitted in the simulated map (Fig. 3). These Spk1 structures have 151 residues, and the total number of free variables was 439. The simulation took ~16 min on the computer described above.

This case provided a different kind of challenge from actin: even though the number of residues was less than half of that of actin, the starting structure was placed purposely with one of its arms in the wrong arm of the density map, and the other arm completely outside the density. (In a real case, one would previously do a rigid-body fitting—using, for instance, CoLoRes (31), FRM (32), URO (33), ADP_EM (34), etc.—and then proceed with the application of DDFF.) Previous methods would get trapped in a local maximum of the correlation function, but our approach was immune to this, and the result of the fitting was again very good: the mismatch with the target structure was mainly in the rotation of the arms, which cannot be captured at this low resolution. Fig. 3, *c* and *d*, show the evolution of the overlap and RMSD with respect to the target: the overlap goes from ~60% to ~99%, while the RMSD goes from 15 Å to ~1.5 Å. Again, these RMSD values include rigid motions.

Tests with experimental maps

Calcium ATPase

The calcium pump, Ca^{2+}-ATPase, is an integral membrane protein that pumps calcium ions across the membrane, relaxing muscle cells (35).

For this test, we used the open conformation of Ca^{2+}-ATPase obtained by Toyoshima et al. (35) at 2.6 Å resolution (PDB accession code 1EUL), and a real EM map at 8 Å resolution previously obtained by Zhang et al. (36). The atomic structure has 994 residues, and the total number of free variables was 2742. This test case is the largest we have considered. The simulation took 235 min on the computer mentioned above.

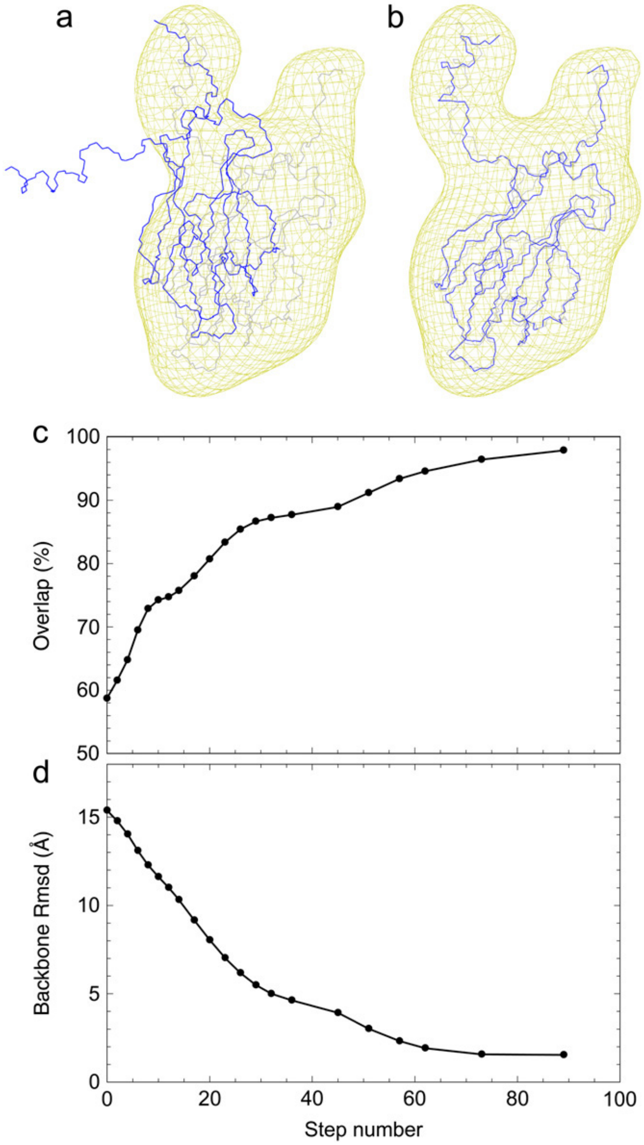

FIGURE 3 Flexible fitting of the open conformation of the Spk1 molecule into a simulated density map (*yellow wireframe*, 1σ above the mean) at 15 Å resolution. Shown in light gray is the closed conformation of Spk1, which was used to generate the map. (*a*) The open conformation of Spk1 is shown in blue. This was the starting conformation used for the fitting. (*b*) The final conformation of the trajectory is displayed in blue. (*c*) Evolution of the overlap values along the trajectory. (*d*) Evolution of the backbone RMSD values along the trajectory.

The atomic structure was initially positioned by eye in the EM map (Fig. 4 *a*). The final conformation (Fig. 4 *b*) had an RMSD of 12.1 Å with respect to the starting one, with a net deformation of 11.2 Å. Fig. 4 *c* shows the evolution of the overlap between the current conformation and the EM map: it goes from 57% to ~76%.

This case was considered earlier by Hinsen et al. (12) using an NMA-based approach, obtaining a result that looks similar to ours, with a similar improvement of the fitting parameter (cross correlation from 70% to 91%) and an almost identical

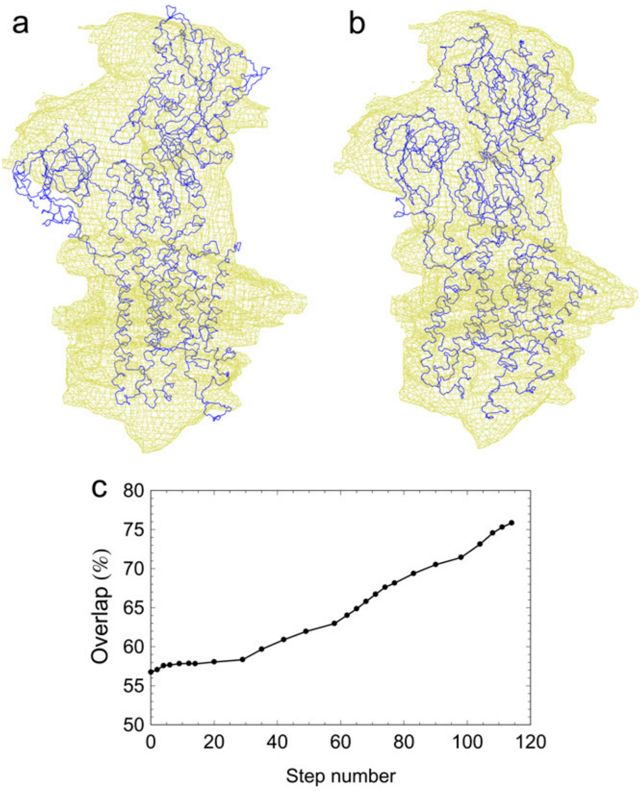

FIGURE 4 Flexible fitting of the open conformation of the Ca^{2+}-ATPase molecule into an experimentally determined density map (*yellow wireframe*, 1σ above the mean) at 8 Å resolution. (*a*) The open conformation of Ca^{2+}-ATPase is shown in blue. This was the starting conformation used for the fitting. (*b*) The final conformation of the trajectory is displayed in blue. (*c*) Evolution of the overlap values along the trajectory.

net deformation (11.3 Å). We chose to give overlap values instead of cross-correlation ones because the overlap (as defined above) depends more linearly on the distance between both maps (hence, it has a better discrimination ability). Instead, the cross correlation tends to give higher values and is more constant as the maps get closer to one another. (A good analogy is to compare the functions $1 - x$ and $\cos x$ as x approaches 0.) However, we remark that we do not use the overlap or any other quantity to perform a minimization—the overlap values are only plotted, but not utilized in any way by the program.

Also, unlike Hinsen et al. (12), we did not see a need (judging by the result obtained) to remove the extra density corresponding to the crystallization agent (decavanadate).

Elongation factor G

EF-G is a member of the GTPase family that catalyzes the translocation step of protein synthesis in bacteria. It consists of five domains, and its structure, in complex with GDP, was initially solved by Czworkowski et al. (37). It was nearly simultaneously solved by Ævarsson et al. (38) without GDP. However, domain III was poorly defined in these structures.

For this test, we used a structure of EF-G containing a point mutation obtained by Laurberg et al. (39) at 2.8 Å resolution (PDB accession code 1FNM), which has all its domains well defined (except for a gap between residues 39 and 68), with a total of 655 residues. We used an EM map at 10.9 Å resolution of EF-G, isolated from a reconstruction of the ribosome with EF-G bound to it (40).

As in the Ca^{2+}-ATPase case, the atomic structure was initially positioned by eye in the EM map (Fig. 5 *a*). But for EF-G we proceeded in two steps. In the first step we fixed all torsion angles within each of the five domains of the molecule, allowing only the linkers between them to be flexible. The segments corresponding to these linkers were based on the definition of the domains given by Ævarsson et al. (38), and consisted of the following 19 residues: 282–287, 401–404, 481–483, 603–605, and 674–676, resulting in only 56 free variables. This step took ~20 min of computing time.

In the second step we allowed full flexibility, and there were 1795 free variables. This part took 46 min of compute time. The transition from the first step to the second is indicated by the black arrow in Fig. 5 *d*, which shows the evolution of the overlap along the trajectory, going from 44% for the initial conformation, through 58% at the transition (Fig. 5 *b*), to 73% for the final conformation (Fig. 5 *c*), which had an RMSD of 8.7 Å with respect to the starting one, with a net deformation of 8.0 Å.

The reason we proceeded in two steps was that there seems to be an unaccounted-for peak of density inside domain I of the EM map, which would attract the other domains toward it. By first rigidifying the domains we were able to alleviate this problem, although not in a perfect way, as is evidenced by the loop on the right side of domain V, which stayed in the neighboring density corresponding to domain I (*red arrow* in Fig. 5 *c*).

Tama et al. (9), by using an NMA-based approach, also performed a flexible fitting of EF-G, obtaining a cross-correlation improvement from 62% to 81%, and a net deformation between the initial and fitted conformations of 8.5 Å. Their use of low-frequency normal modes avoided the small displacement of the arrowed loop in this particular case. However, their timing was 5 h on a Xeon 2.4 GHz processor.

Transition pathways

One of the potential applications of DDFF that we envision is the computation of transition pathways between two given conformers of a molecule. This application is very straightforward to implement: it involves only a modification of the definition of the force field acting on the atomic structure. Namely, each atom of the origin structure is attracted by the corresponding atom of the target structure, with a force that is proportional to the distance between both atoms. For simplicity, we have defined the force field only on the C_α atoms.

We have applied this to the case of Adenylate Kinase (ADK), an enzyme that catalyzes the transfer of a phosphoryl group in a reaction involving ATP and AMP (41). We used an atomic structure of ADK in its open conformation (PDB accession code 4AKE) as the starting conformer, and a structure in a closed conformation (PDB 1AKE) as the target conformer (Fig. 6). These structures have 214 residues, and the number of free variables was 587. The simulation took only 2 min, evidencing that most of the computing time in the foregoing cases was spent in map-related computations.

Fig. 6 *d* shows the evolution of the C_α RMSD of each conformation along the trajectory with respect to the target: it goes from 7.1 Å to 0.2 Å. An intermediate conformation (step 19 in the plot, with 4 Å RMSD relative to the target) and the last one are shown in Fig. 6, *b* and *c*. The slight differences between the last conformation and the target (accounting for the 0.2 Å RMSD) are due to the fact that only the C_α atoms were subjected to the force field.

ADK has been used by many authors as a test case. A recent work (42) used a combination of NMA and a robotics-like path planning algorithm to compute the transition pathway between the above two conformers. Although their trajectory looks very reasonable, the final conformation is still ~2 Å RMSD from the target, and takes considerable longer computing time (80 min).

Another work that dealt with conformational changes of ADK used a so-called plastic network model (43). This approach consists in considering a harmonic energy basin around each of the endpoint conformers, say $G_1(\mathbf{x})$ and $G_2(\mathbf{x})$, and in defining a smooth version of their pointwise minimum $G(\mathbf{x}) = \min\{G_1(\mathbf{x}), G_2(\mathbf{x})\}$. Then a path of least G-action (i.e., a mountain-pass path through the saddle point) is computed from the first conformer to the second. Their results are excellent: the RMSD between the last conformer of the trajectory and the target is negligible (<0.1 Å, according to their plot), and all the intermediate conformers are within 3 Å from at least one experimental structure of ADK.

Homology and loop modeling

The second potential application that we propose DDFF could be used for is homology modeling. This is essentially no different, as far as the code is concerned, from transition pathways. We again have two molecules with some sequence identity, and we use the target one to pull the other according to a given residue-correspondence table—presumably obtained by aligning both sequences.

As a simple example, we applied this to the modeling of a different conformation of human SIRT2 (a homolog of yeast

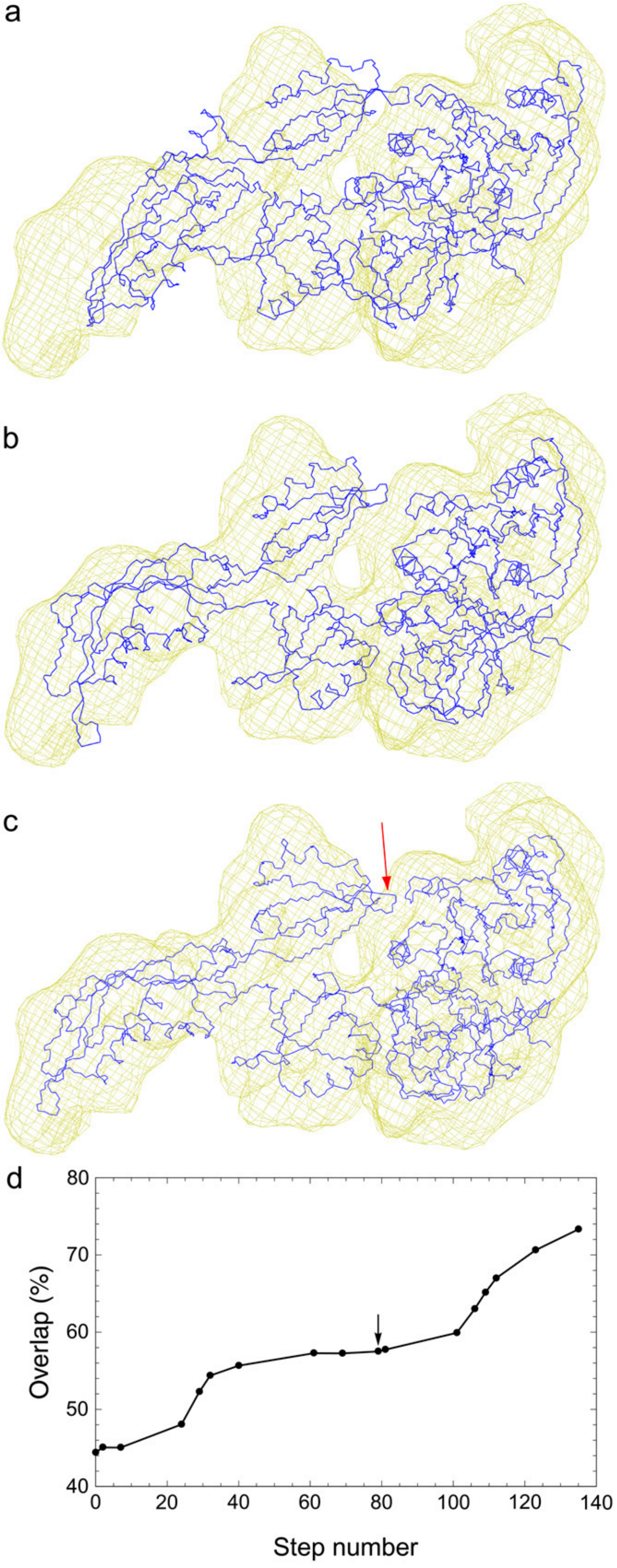

FIGURE 5 Flexible fitting of the compact conformation of the EF-G molecule into an experimentally determined density map (*yellow wireframe*, 1σ above the mean) of an extended conformation at 10.9 Å resolution. (*a*) The compact conformation of EF-G is shown in blue. This was the starting conformation used for the fitting. (*b*) Conformation obtained by keeping the five domains of EF-G rigid and allowing only the linkers between them to be flexible. (*c*) Final conformation (in *blue*) of the trajectory obtained by starting with the conformation in *b* and allowing full flexibility. The red arrow indicates the loop mentioned in the text, which we believe should go into the density of domain V rather than that of domain I. (*d*) Evolution of the overlap values along the trajectory. The black arrow indicates the point where the transition from rigid domains to flexible domains occurred.

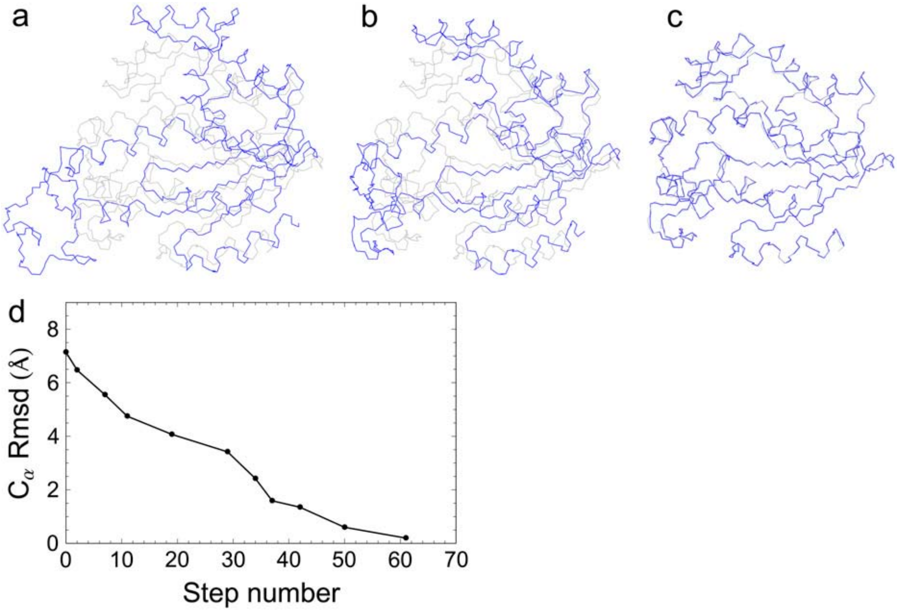

FIGURE 6 Flexible superposition of the open conformation of the ADK molecule onto the closed conformation (*gray wire*). (*a*) The open conformation of ADK is shown in blue. (*b*) A midway conformation (step 19 in *d*) is displayed in blue. It has a 4 Å RMSD relative to the target conformation. (*c*) The final conformation of the trajectory is displayed in blue. The slight difference with the target conformation is due to that fact that the force field was defined only on the C_α atoms. (*d*) Evolution of the C_α-RMSD values along the trajectory.

Sir2) from a known structure (44) (PDB accession code 1J8F), using as a template (target) the structure of yeast Hst2 (45) (PDB 1SZD). These enzymes are believed to play roles in gene silencing, DNA repair, genome stability, longevity, and metabolism.

These structures, shown in Fig. 7, have 300 and 296 residues, respectively, although there are some insertions/deletions in the alignment, and the segment 210–214 in the target structure is missing—disordered according to Zhao et al. (45). The number of free variables in the simulation was 831, and it took 2 min of compute time.

Fig. 7 *c* shows the evolution of the C_α RMSD of each conformation along the trajectory with respect to the target: it goes from 5.1 Å to 0.9 Å, although the loops at the lower left of Fig. 7, *a* and *b*, move much more, ~11 Å (there is an important component of the motion perpendicular to the plane of the article), while the core domain on the right of the molecule stays virtually still. (We do not know the cause of the peak at step 44 of the plot.) The loop at the lower right of Fig. 7, *a* and *b*, is where the missing segment 210–214 of Hst2 should be (replaced by a straight line segment), and is also the place of an important insertion of Hst2 relative to SIRT2. This, and other insertions/deletions, are easily visible in Fig. 7 *b* as regions where there is lack of superposition between the last conformer of the trajectory and the target.

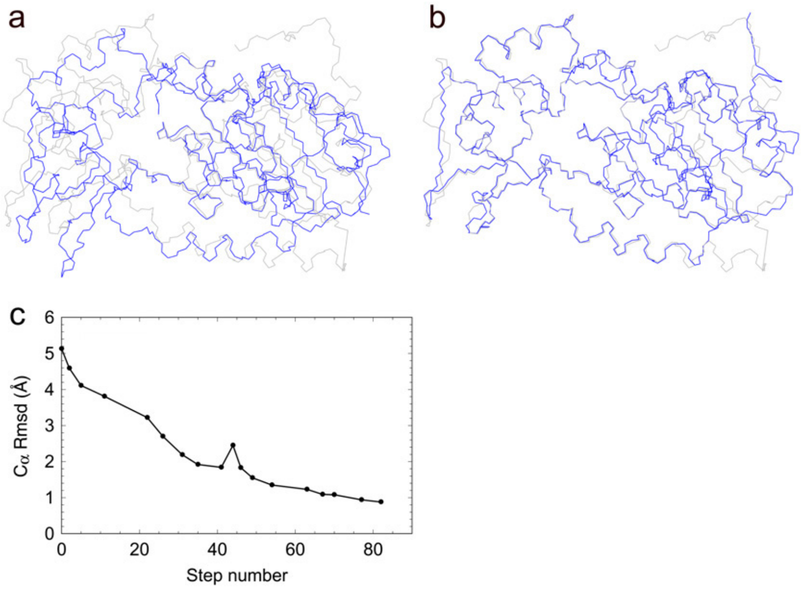

FIGURE 7 Flexible superposition of the SIRT2 molecule onto the Hst2 molecule (*gray wire*). (*a*) SIRT2 is shown in blue. (*b*) The final conformation of the trajectory is displayed in blue. The alignment of these two molecules contains some insertions and deletions, resulting in corresponding stretches where there is no superposition (in addition to the slight differences everywhere due to the force field being defined, as in the ADK case, only on the C_α atoms). (*c*) Evolution of the C_α-RMSD values along the trajectory. The comparatively large final value (0.88 Å) is due to the insertions and deletions just mentioned.

DISCUSSION AND CONCLUSIONS

We developed a novel approach, DDFF, to perform flexible fitting of atomic structures into EM maps that works entirely in internal coordinates q_i (torsion angles and global position of each chain), thus preserving rigorously the covalent structure of the molecule. It allows the modeling of arbitrarily large conformational changes, using dampers to maintain the overall assembly of the molecule, especially its secondary-structure elements. The structure is evolved according to a force field produced by a given EM map. The system is made completely damped by setting all masses to 0. As a consequence, the equations of motion are of first-order and linear in the $\dot{q}_i$, allowing for a much simpler and efficient numerical integration scheme.

We validated our approach on two cases with simulated EM maps: Actin and Spk1. For each, one of the conformations was fitted in a map generated by the other one. The final RMSD were, in both cases, excellent: between 1 Å and 1.5 Å for the backbone atoms.

We have applied DDFF to two experimentally obtained EM maps: Elongation Factor G and Ca^{2+}-ATPase. The overlaps between the final conformations and the EM maps were very good: 73–76%. The timings were attractive as well: 1 h for EF-G (1795 free variables) and 4 h for Ca^{2+}-ATPase (2742 free variables) on a modest 1.1-GHz AMD Opteron workstation under Linux. This is evidence that our method scales well with the system's size, due mainly to the $O(N^2)$ complexity of the computation of the system matrix.

We also proposed two potential applications of our approach: transition pathways and homology/loop modeling. These involve straightforward modifications of the force field that pulls the atomic structure: instead of being generated by a map, the forces are defined simply to be proportional to the distance between each atom in the origin structure and the corresponding atom in the target structure. As an example of transition pathways, we considered ADK, obtaining a transition between its open and its closed conformations. These structures had 587 free variables and the calculation was done in only 2 min.

As an example of homology modeling, we obtained a different conformation of human SIRT2 from a known structure, using as a template (target) the structure of yeast Hst2. The number of free variables in this case was 831, and it also took ~2 min of compute time.

It is also possible to define the force field to handle protein-protein docking, taking into account the flexibility of both partners. The basic idea is to consider the molecules to be docked as two chains of a single atomic structure, and the force field is defined between certain subsets of the surface atoms of each chain (Fig. 8). The figure intends to represent one of a set of relative positions of the molecules that have been obtained after an initial rigid-body search. This search could be either exhaustive or, better still, based on an interface prediction methodology such as PIER (46), which predicts which patches on each protein's surface (independently of one another) are likely to be docking interfaces. These interfaces are indicated in Fig. 8 by the thick lines *A* and *B*. The force field is then defined only on these interfaces, so that each atom in face *A* is attracted by all atoms in face *B*, and vice versa. The magnitude of the force exerted by an atom is proportional to its minimum distance to any atom of the opposing face. The final conformation of the DDFF trajectory is shown in the right side of Fig. 8.

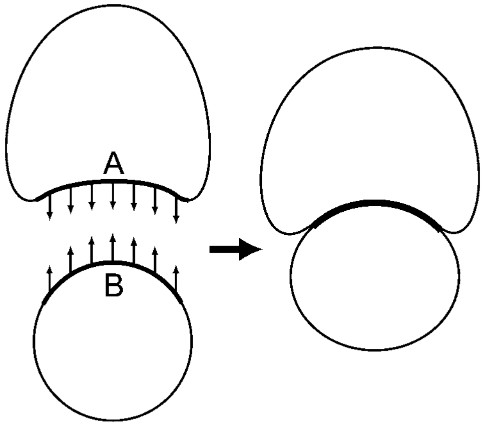

FIGURE 8 Illustration of the application of DDFF to protein-protein docking. The force field is defined only on surface atoms (thick faces *A* and *B*), so that each atom in face *A* is attracted by all atoms in face *B*, and vice versa. The magnitude of the force exerted by an atom is proportional to the minimum distance to any atom of the opposing face.

A word regarding overfitting. The way we controlled this was as follows. Our simulations were allowed to run until the convergence criterion was met. Then we inspected the output log and noticed a feature common to all cases: a clear-cut point along the trajectory at which the RMSD begins to decrease markedly. (Recall that the RMSD is updated at each step according to the current and previous velocity fields.) That point was taken as the final conformation shown in our examples. We interpret that particular point as the point where overfitting begins to occur. This is especially clear in the cases of Ca^{2+}-ATPase and EF-G, for which real EM maps were used. We thus have an effective way to find the optimal fitted conformation that is free from overfitting.

Finally, we claim that our method is more immune to local minima—or points where the force is 0—than previous approaches are, in particular that of Hinsen et al. (12), to which ours has certain similarity. The reason becomes clear upon comparing the expressions of the forces used by these methods.

Hinsen's method (12):

$$\mathbf{F}_k^{(m)} = -\int \{f(\mathbf{p}) - g(\mathbf{p};\mathbf{r}_1,\ldots,\mathbf{r}_N)\} \cdot \frac{\partial g}{\partial \mathbf{r}_k} d\mathbf{p}.$$

Our approach (from Eq. 13, omitting terms irrelevant to this comparison):

$$\mathbf{F}_k^{(m)} = -\int \{f(\mathbf{p}) - g(\mathbf{p};\mathbf{r}_1,\ldots,\mathbf{r}_N)\} \cdot (\mathbf{r}_k - \mathbf{p}) d\mathbf{p}.$$

The factor $(\partial g/\partial \mathbf{r}_k)$ in Hinsen's approach could easily vanish (for instance, in the Spk1 case, see Fig. 3). Instead, our approach has the factor $\mathbf{r}_k - \mathbf{p}$, which produces a force toward (or away from, according to the sign of $f - g$) points of the EM map f whose densities are different from those of the current structure-induced map g.

However, our approach is not completely immune to local minima. We have not implemented safeguards against them thus far, but we envision that one may make the force-field variable over time to overcome them, and also one could periodically perform a Fast Rotational Matching step (32) to improve the orientation and help the escape from such critical points. It might also be necessary to perform the fitting process starting from a number of initial positions and orientations.

APPENDIX A: COMPUTATION OF THE MATRICES V AND B

Recursive formulas for the V matrix

Recall that the generalized forces produced by the dampers are given by $Q_j^{(s)} = \sum_k \langle \mathbf{F}_k^{(s)}, (\partial \mathbf{r}_k/\partial q_j)\rangle = -\sum_i V_{ij}\dot{q}_i$, where the V_{ij} value can be written in the compact form as

$$V_{ij} = \sum_{k<l} C_{kl} \cdot \frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_i} \cdot \frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_j}. \tag{16}$$

The problem with the direct application of this expression is that it takes $O(M^2N^2)$ operations to compute all of the V_{ij} values. (M is the number of free variables and N is the number of atoms.) We were able to avoid this serious bottleneck by rewriting Eq. 16 in a way amenable to recursive computation, which is formally the same as Eq. 10 in Abe et al. (24), thereby reducing the total number of operations to $O(M^2) + O(N^2)$. In the usual case where all (or most) of the variables are free, it is $O(M) = O(N)$, hence the complexity reduction achieved in this way is from $O(N^4)$ to $O(N^2)$.

We start by writing Eq. 16 as

$$V_{ij} = \sum_{k<l} C_{kl}^{(j)} \cdot \frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_i}, \tag{17}$$

where

$$C_{kl}^{(j)} = C_{kl} \frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_j}.$$

We have

$$\frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_i} = \begin{cases} 0 & \text{if} \quad k \in M_i \text{ and } l \in M_i \\ & \text{or} \quad k \in \overline{M}_i \text{ and } l \in \overline{M}_i, \\ \left\langle \dfrac{\partial \mathbf{r}_l}{\partial q_i}, \dfrac{\mathbf{r}_l - \mathbf{r}_k}{\|\mathbf{r}_l - \mathbf{r}_k\|} \right\rangle & \text{if} \quad k \in \overline{M}_i \text{ and } l \in M_i, \\ \left\langle \dfrac{\partial \mathbf{r}_k}{\partial q_i}, \dfrac{\mathbf{r}_k - \mathbf{r}_l}{\|\mathbf{r}_k - \mathbf{r}_l\|} \right\rangle & \text{if} \quad k \in M_i \text{ and } l \in \overline{M}_i, \end{cases} \tag{18}$$

where M_i is the set of atoms that are actually moved by q_i,

$$M_i = \{k | \mathbf{r}_k \text{ depends on } q_i\}.$$

Using these expressions in Eq. 17, we get

$$V_{ij} = \sum_{\substack{k \in M_i \\ l \in \overline{M}_i}} \tilde{C}_{kl}^{(j)} \left\langle \frac{\partial \mathbf{r}_k}{\partial q_i}, \mathbf{r}_k - \mathbf{r}_l \right\rangle, \tag{19}$$

where

$$\tilde{C}_{kl}^{(j)} = \frac{C_{kl}^{(j)}}{\|\mathbf{r}_k - \mathbf{r}_l\|} = \frac{C_{kl}}{\|\mathbf{r}_k - \mathbf{r}_l\|} \cdot \frac{\partial \|\mathbf{r}_k - \mathbf{r}_l\|}{\partial q_j}.$$

We can write down explicit formulas for the derivatives of a position with respect to the generalized variables. These will depend upon the type of generalized variable: either angular (torsion angles) or Cartesian (coordinates x, y, z of the first atom of a chain). For $k \in M_i$, we have the equations

$$\frac{\partial \mathbf{r}_k}{\partial q_i} = \begin{cases} \mathbf{e}_i \wedge (\mathbf{r}_k - \mathbf{r}_{\beta(i)}) & \text{if} \quad q_i \text{ is angular,} \\ \mathbf{u}_\xi & \text{if} \quad q_i = x^\xi, \end{cases} \tag{20}$$

where $\mathbf{e}_i$ is the rotation axis associated to q_i, and $\mathbf{r}_{\beta(i)}$ is a point on that axis. "$\wedge$" denotes the vector product in $\mathbb{R}^3$. The index ξ can assume values 1, 2, and 3; x^1, x^2, and x^3 denote the Cartesian coordinates x, y, and z; and $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ denote the unit vectors (1, 0, 0), (0, 1, 0), and (0, 0, 1).

If q_i is an angular variable, we get, upon substituting the first of the expressions in Eq. 20 in Eq. 19 and making a few rearrangements,

$$V_{ij} = -\langle \mathbf{e}_i, F_{ij} \rangle - \langle \mathbf{e}_i \wedge \mathbf{r}_{\beta(i)}, G_{ij} \rangle,$$

where

$$F_{ij} = \sum_{\substack{k \in M_i \\ l \in \bar{M}_i}} \tilde{C}_{kl}^{(j)} (\mathbf{r}_k \wedge \mathbf{r}_l), \quad G_{ij} = \sum_{\substack{k \in M_i \\ l \in \bar{M}_i}} \tilde{C}_{kl}^{(j)} (\mathbf{r}_k - \mathbf{r}_l). \tag{21}$$

Equation 21 is analogous to Eq. 5 in Abe et al. (24), except for the presence of the index j in the coefficients of F_{ij} and G_{ij}, which would produce, in our case, a complexity of $O(M^3)$ if the recursive approach were applied directly to Eq. 21. Therefore, to get recursion formulas with $O(M^2)$ complexity, we go further and substitute Eq. 18 into the above formulas for F_{ij} and G_{ij}. For the former, we get

$$\begin{aligned} F_{ij} = &- \sum_{\substack{k \in \bar{M}_i \cap M_j \\ l \in M_i \cap \bar{M}_j}} \frac{C_{kl}}{\|\mathbf{r}_k - \mathbf{r}_l\|^2} \cdot \left\langle \frac{\partial \mathbf{r}_k}{\partial q_j}, \mathbf{r}_k - \mathbf{r}_l \right\rangle \cdot (\mathbf{r}_k \wedge \mathbf{r}_l) \\ &+ \sum_{\substack{k \in M_i \cap M_j \\ l \in \bar{M}_i \cap \bar{M}_j}} \frac{C_{kl}}{\|\mathbf{r}_k - \mathbf{r}_l\|^2} \cdot \left\langle \frac{\partial \mathbf{r}_k}{\partial q_j}, \mathbf{r}_k - \mathbf{r}_l \right\rangle \cdot (\mathbf{r}_k \wedge \mathbf{r}_l). \end{aligned}$$

According to how the sets M_i and M_j relate to one another, three cases can be distinguished: Case 1, $M_i \cap M_j = \emptyset$; Case 2, $M_i \subset M_j$; and Case 3, $M_j \subset M_i$. No other case is possible, because it is easy to see that if $M_i \cap M_j \neq \emptyset$, then one of them must be contained in the other. In Case 1, $M_j \subset \overline{M}_i$, so the second term is absent. Similarly, in Cases 2 and 3, the first term is absent. Hence, we can write

$$F_{ij} = \sigma_{ij} \sum_{(k,l) \in P_{ij}} D_{kl} \cdot \left\langle \frac{\partial \mathbf{r}_k}{\partial q_j}, \mathbf{r}_k - \mathbf{r}_l \right\rangle \cdot (\mathbf{r}_k \wedge \mathbf{r}_l), \tag{22}$$

where

$$D_{kl} = \frac{C_{kl}}{\|\mathbf{r}_k - \mathbf{r}_l\|^2},$$

$$\sigma_{ij} = \begin{cases} -1 & \text{if} \quad M_i \cap M_j = \emptyset, \\ +1 & \text{if} \quad M_i \subset M_j \text{ or } M_j \subset M_i, \end{cases}$$

$$P_{ij} = \begin{cases} M_j \times M_i & \text{if} \quad M_i \cap M_j = \emptyset, \\ M_i \times \overline{M}_j & \text{if} \quad M_i \subset M_j, \\ M_j \times \overline{M}_i & \text{if} \quad M_j \subset M_i. \end{cases}$$

The corresponding expression for G_{ij} is analogous, the only difference being in replacing $\mathbf{r}_k \wedge \mathbf{r}_l$ by $\mathbf{r}_k - \mathbf{r}_l$:

$$G_{ij} = \sigma_{ij} \sum_{(k,l)\in P_{ij}} D_{kl} \cdot \left\langle \frac{\partial \mathbf{r}_k}{\partial q_j}, \mathbf{r}_k - \mathbf{r}_l \right\rangle \cdot (\mathbf{r}_k - \mathbf{r}_l). \tag{23}$$

Recall that we are assuming that q_i is an angular variable. To proceed from here, we have to distinguish two cases: q_j can be either angular or Cartesian. If q_j is angular, we substitute the first of the expressions in Eq. 20 into Eqs. 22 and 23, and the resulting expressions into Eq. 21, obtaining

$$\begin{aligned}\sigma_{ij}V_{ij} = \sum_{(k,l)\in P_{ij}} D_{kl}[&\langle \mathbf{e}_i, \mathbf{r}_k \wedge \mathbf{r}_l\rangle \cdot \langle \mathbf{e}_j, \mathbf{r}_k \wedge \mathbf{r}_l\rangle + \langle \mathbf{e}_i, \mathbf{r}_k \wedge \mathbf{r}_l\rangle \\ &\cdot \langle \mathbf{e}_j \wedge \mathbf{r}_{\beta(j)}, \mathbf{r}_k - \mathbf{r}_l\rangle + \langle \mathbf{e}_j, \mathbf{r}_k \wedge \mathbf{r}_l\rangle \cdot \langle \mathbf{e}_i \wedge \mathbf{r}_{\beta(i)}, \mathbf{r}_k - \mathbf{r}_l\rangle \\ &+ \langle \mathbf{e}_i \wedge \mathbf{r}_{\beta(i)}, \mathbf{r}_k - \mathbf{r}_l\rangle \cdot \langle \mathbf{e}_j \wedge \mathbf{r}_{\beta(j)}, \mathbf{r}_k - \mathbf{r}_l\rangle].\end{aligned}$$

We compute these scalar products by taking components, denoted below by the superindices μ and ν. We get

$$\begin{aligned}\sigma_{ij}V_{ij} = \sum_{\mu,\nu=1}^{3} [&U_{ij}^{\mu\nu}\mathbf{e}_i^\mu \mathbf{e}_j^\nu + H_{ij}^{\mu\nu}\mathbf{e}_i^\mu(\mathbf{e}_j \wedge \mathbf{r}_{\beta(j)})^\nu \\ &+ H_{ij}^{\nu\mu}\mathbf{e}_j^\nu(\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^\mu \\ &+ W_{ij}^{\mu\nu}(\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^\mu(\mathbf{e}_j \wedge \mathbf{r}_{\beta(j)})^\nu],\end{aligned}$$

where

$$\left.\begin{aligned}U_{ij}^{\mu\nu} &= \sum_{(k,l)\in P_{ij}} D_{kl} \cdot (\mathbf{r}_k \wedge \mathbf{r}_l)^\mu \cdot (\mathbf{r}_k \wedge \mathbf{r}_l)^\nu, \\ H_{ij}^{\mu\nu} &= \sum_{(k,l)\in P_{ij}} D_{kl} \cdot (\mathbf{r}_k \wedge \mathbf{r}_l)^\mu \cdot (\mathbf{r}_k - \mathbf{r}_l)^\nu, \\ W_{ij}^{\mu\nu} &= \sum_{(k,l)\in P_{ij}} D_{kl} \cdot (\mathbf{r}_k - \mathbf{r}_l)^\mu \cdot (\mathbf{r}_k - \mathbf{r}_l)^\nu.\end{aligned}\right\} \tag{24}$$

This can be conveniently written in matrix form (all vectors are considered as columns, rather than rows, in this product; superscript T denotes the transpose of a matrix),

$$V_{ij} = \sigma_{ij}(\mathbf{e}_i^T, (\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^T) \cdot R_{ij} \cdot \begin{pmatrix} \mathbf{e}_j \\ \mathbf{e}_j \wedge \mathbf{r}_{\beta(j)} \end{pmatrix}, \tag{25}$$

where

$$R_{ij} = \begin{pmatrix} U_{ij} & H_{ij} \\ H_{ij}^T & W_{ij} \end{pmatrix}, \tag{26}$$

a 6 × 6 symmetric matrix.

The key point in writing V_{ij} in the above manner is that the variable indices i, j are separated from the atom indices k, l, since, in view of the expressions for the U_{ij}, H_{ij}, and W_{ij} matrices, the matrix R_{ij} can be written as

$$R_{ij} = \sum_{(k,l)\in P_{ij}} L_{kl},$$

where the 6 × 6 matrices L_{kl} do not depend on i, j. This expression of the R_{ij} makes it possible to compute them recursively. This is so because the sets M_i form an increasing sequence as one goes backward along the chain, and likewise, the complements $\overline{M}_i$ form an increasing sequence as one goes forward along the chain. For example, we have, if R denotes the residue index,

$$\begin{aligned}M_{\phi_R} &= V_{\phi_R} \cup M_{\psi_R} \cup M_{\chi_R} \\ \overline{M}_{\phi_R} &= V_{\psi_{R-1}} \cup \overline{M}_{\psi_{R-1}},\end{aligned}$$

where V_i is the set of atoms that depend only on q_i when the variables before q_i are kept fixed. These units V_i (for i ranging over all free variables) thus form a partition of the set of all atoms.

Equation 26 is analogous to Eq. 10 in Abe et al. (24). Details of the recursive algorithm for the computation of the R_{ij} are given in that reference.

Let us now consider the case in which q_i is an angular variable and q_j is a Cartesian variable, say $q_j = x^\xi$. We substitute the second of the expressions in Eq. 20 into Eqs. 22 and 23, and the resulting expressions into Eq. 21, obtaining, after the introduction of components as before,

$$\sigma_{ij}V_{ij} = -\sum_{\mu=1}^{3}\left[\mathbf{e}_i^\mu H_{ij}^{\mu\xi} + (\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^\mu W_{ij}^{\mu\xi}\right],$$

which can be written in matrix form,

$$\sigma_{ij}V_{ij} = -(\mathbf{e}_i^T, (\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^T) \cdot \mathrm{col}_{\xi+3}(R_{ij}), \tag{27}$$

where col_n signifies the n^{th} column of the matrix.

If q_i is Cartesian, say $q_i = x^\xi$, the computation starts with Eq. 19, in which we substitute the second of the expressions in Eq. 20, obtaining

$$V_{ij} = \langle \boldsymbol{u}_\xi, G_{ij}\rangle. \tag{28}$$

We now distinguish two cases: q_j either angular or Cartesian.

If q_j is angular, we use the first of Eqs. 20 into Eq. 23, and substitute the resulting expression into Eq. 28, obtaining, after taking components as before,

$$\sigma_{ij}V_{ij} = -\sum_{\mu=1}^{3}\left[\mathbf{e}_j^\mu H_{ij}^{\mu\xi} + (\mathbf{e}_j \wedge \mathbf{r}_{\beta(j)})^\mu W_{ij}^{\mu\xi}\right],$$

which can be written in matrix form as

$$\sigma_{ij}V_{ij} = -(\mathbf{e}_j^T, (\mathbf{e}_j \wedge \mathbf{r}_{\beta(j)})^T) \cdot \mathrm{col}_{\xi+3}(R_{ij}). \tag{29}$$

Finally, if q_j is Cartesian, say $q_j = x^\eta$, we use the second of the expressions in Eq. 20 in Eq. 23, and substitute the resulting expression into Eq. 28, obtaining, after taking components as before,

$$\sigma_{ij}V_{ij} = W_{ij}^{\xi\eta} = R_{ij}^{\xi+3,\eta+3}. \tag{30}$$

Recursive formulas for the B matrix

Recall that the generalized forces produced by the water resistance are given by $Q_j^{(d)} = \sum_k\langle \mathbf{F}_k^{(d)}, (\partial \mathbf{r}_k/\partial q_j)\rangle = -\sum_i B_{ij}\dot{q}_i$, where the B_{ij} values are given by Eq. 10. Since terms for which k does not belong to M_i or M_j vanish, the summation reduces to

$$B_{ij} = \sum_{k\in M_i \cap M_j} b_k \left\langle \frac{\partial \mathbf{r}_k}{\partial q_i}, \frac{\partial \mathbf{r}_k}{\partial q_j} \right\rangle. \tag{31}$$

Similarly to what happens with the $\mathbf{V}$ matrix, the direct application of this expression would take $O(M^2N)$ operations (where M is the number of free variables and N is the number of atoms). However, it is possible to perform, as with the $\mathbf{V}$ matrix, a separation of indices (the atom indices k, l from the variable indices i, j) that yields recursion formulas that allow the computation of all the B_{ij} in $O(M^2) + O(N)$ operations. In the case that (almost) all of the variables are free, $O(M) = O(N)$ and the complexity reduction is from $O(N^3)$ to $O(N^2)$.

We start by considering the case in which both q_i and q_j are angular variables. We substitute the first of the expressions in Eq. 20 into Eq. 31, and by using the identity

$$\langle A \wedge B, C \wedge D\rangle = \langle A, C\rangle\langle B, D\rangle - \langle B, C\rangle\langle A, D\rangle,$$

we arrive, after a short calculation, at

$$\begin{aligned} B_{ij} = \sum_{k \in M_i \cap M_j} \big[& \langle \mathbf{e}_i, \mathbf{e}_j \rangle \langle \mathbf{r}_k, \mathbf{r}_k \rangle - \langle \mathbf{e}_j, \mathbf{r}_k \rangle \langle \mathbf{e}_i, \mathbf{r}_k \rangle - \langle \mathbf{e}_i, \mathbf{e}_j \rangle \langle \mathbf{r}_k, \mathbf{r}_{\beta(j)} \rangle \\ & + \langle \mathbf{e}_j, \mathbf{r}_k \rangle \langle \mathbf{e}_i, \mathbf{r}_{\beta(j)} \rangle - \langle \mathbf{e}_i, \mathbf{e}_j \rangle \langle \mathbf{r}_{\beta(i)}, \mathbf{r}_k \rangle + \langle \mathbf{e}_j, \mathbf{r}_{\beta(i)} \rangle \langle \mathbf{e}_i, \mathbf{r}_k \rangle \\ & + \langle \mathbf{e}_i \wedge \mathbf{r}_{\beta(i)}, \mathbf{e}_j \wedge \mathbf{r}_{\beta(j)} \rangle \big]. \end{aligned}$$

By computing the scalar products through components, we obtain

$$\begin{aligned} B_{ij} = {} & \langle \mathbf{e}_i, \mathbf{e}_j \rangle \cdot \mathrm{Tr}(S_{ij}) - \mathbf{e}_i^{\mathrm{T}} \cdot S_{ij} \cdot \mathbf{e}_j + \langle \mathbf{e}_i \wedge \mathbf{r}_{\beta(i)}, \mathbf{e}_j \wedge \mathbf{r}_{\beta(j)} \rangle A_{ij} \\ & + \langle \langle \mathbf{e}_i, \mathbf{r}_{\beta(j)} \rangle \mathbf{e}_j + \langle \mathbf{e}_j, \mathbf{r}_{\beta(i)} \rangle \mathbf{e}_i - \langle \mathbf{e}_i, \mathbf{e}_j \rangle (\mathbf{r}_{\beta(i)} + \mathbf{r}_{\beta(j)}), E_{ij} \rangle, \end{aligned} \tag{32}$$

where *Tr* denotes the trace of a matrix (sum of the diagonal entries) and

$$\left. \begin{aligned} S_{ij}^{\mu\nu} &= \sum_{k \in M_i \cap M_j} b_k \mathbf{r}_k^{\mu} \mathbf{r}_k^{\nu}, \\ E_{ij} &= \sum_{k \in M_i \cap M_j} b_k \mathbf{r}_k, \\ A_{ij} &= \sum_{k \in M_i \cap M_j} b_k. \end{aligned} \right\}. \tag{33}$$

These quantities can be combined into a single 10-dimensional vector Z_{ij} (since S_{ij} is a symmetric matrix), and the expressions in Eq. 20 can therefore be written as

$$Z_{ij} = \sum_{k \in M_i \cap M_j} Y_k,$$

where the 10-dimensional vectors Y_k do not depend on i, j. As before, this is the key for obtaining a recursive algorithm for the B_{ij}.

Now, for pairs i, j that give $M_i \cap M_j = \emptyset$, we automatically get $B_{ij} = 0$. Otherwise, we have either $M_i \subset M_j$ or $M_j \subset M_i$. Since (from Eq. 18) $B_{ij} = B_{ji}$, we need only consider, say, the second case, which implies $M_i \cap M_j = M_j$. Hence, Z_{ij} reduces to

$$Z_j = \sum_{k \in M_j} Y_k.$$

This expression for Z_j makes it very easy to compute them recursively, since the sets M_j form an increasing sequence as one goes backward along the chain. For example, suppose $q_j = \psi_R(A)$ (meaning residue R of chain A). Then $M_{\psi_R(A)} = V_{\psi_R(A)} \cup M_{\phi_{R+1}(A)}$. (Recall that V_i denotes the set of atoms that depend only on q_i when the variables before q_i are kept fixed.) Hence the recursive equation for Z_j is

$$Z_{\psi_R(A)} = t_{\psi_R(A)} + Z_{\phi_{R+1}(A)},$$

where

$$t_{\psi_R(A)} = \sum_{k \in V_{\psi_R(A)}} Y_k.$$

Let us now consider the case in which q_i is an angular variable and q_j is a Cartesian variable, say $q_j = x^{\xi}$. Upon substituting the expressions in Eq. 20 into Eq. 31, we arrive, after a short calculation, at

$$B_{ij} = (\mathbf{e}_i \wedge E_{ij})^{\xi} - (\mathbf{e}_i \wedge \mathbf{r}_{\beta(i)})^{\xi} A_{ij}. \tag{34}$$

The case in which $q_i = x^{\xi}$ and q_j is angular is analogous—just a reversal of the indices

$$B_{ij} = (\mathbf{e}_j \wedge E_{ij})^{\xi} - (\mathbf{e}_j \wedge \mathbf{r}_{\beta(j)})^{\xi} A_{ij}. \tag{35}$$

Finally, if both variables are Cartesian, say $q_i = x^{\xi}$ and $q_j = x^{\eta}$, we immediately get

$$B_{ij} = \delta_{\xi\eta} A_{ij}, \tag{36}$$

where $\delta_{\xi\eta} = 1$ if $\xi = \eta$ and 0 otherwise.

APPENDIX B: ONE-ATOM EXAMPLE

Here we consider the simplest possible system: a single atom to be fitted into its density map. The purpose of doing this is to know the qualitative nature of the trajectory and its rate of convergence. The density produced by the atom is modeled as a Gaussian centered at the origin

$$f(\mathbf{p}) = e^{-\frac{\|\mathbf{p}\|^2}{2\sigma^2}}.$$

Due to the symmetry of this system, the atom will move on a straight line, which for convenience is assumed to be the z axis. Thus, the only generalized variable is $q_1 = z$, and the position vector of the atom is $\mathbf{r}_1 = (0, 0, z)$. The Wilson's matrix consists of the single vector

$$\frac{\partial \mathbf{r}_1}{\partial q_1} = \frac{\partial \mathbf{r}_1}{\partial z} = (0, 0, 1),$$

and the matrices **B** and **V** reduce to single numbers,

$$\begin{aligned} B_{11} &= b_1 \left\langle \frac{\partial \mathbf{r}_1}{\partial q_1}, \frac{\partial \mathbf{r}_1}{\partial q_1} \right\rangle = b_1, \\ V_{11} &= 0\ (\text{since the sum is empty}). \end{aligned}$$

Therefore, the equation of motion is

$$\dot{z} = \frac{W}{b_1}, \tag{37}$$

where

$$W = W(z) = Q_1^{(m)} = \left\langle \mathbf{F}_1^{(m)}, \frac{\partial \mathbf{r}_1}{\partial q_1} \right\rangle = (\mathbf{F}_1^{(m)})_3$$

is the only non-null component of the force. (The subscript *3* means the third component of the vector.)

The structure-generated density is

$$g(\mathbf{p}, z) = e^{-\frac{\|\mathbf{p}-(0,0,z)\|^2}{2\sigma^2}} = f(\mathbf{p} - (0, 0, z)),$$

and, according to Eq. 13, upon making the substitution $\mathbf{p} - (0, 0, z) = \mathbf{v}$, the force W can be written as (assuming for simplicity that the constant in front is *1*):

$$W(z) = \int_{\mathbb{R}^3} (f(\mathbf{v} + (0, 0, z)) - f(\mathbf{v})) \cdot A(\|\mathbf{v}\|) \cdot v_3 \cdot d\mathbf{v}.$$

To be able to solve Eq. 37, we will use only a linear approximation of $W(z)$ near $z = 0$, since we are interested in the behavior of just the final part of the trajectory. Since $W(0) = 0$, we have $W(z) \approx W'(0)z$ for small z. The derivative $W'(0)$ is given by

$$W'(0) = \int_{\mathbb{R}^3} \partial_3 f(\mathbf{v}) \cdot A(\|\mathbf{v}\|) \cdot v_3 \cdot d\mathbf{v},$$

where $\partial_3 f$ denotes the partial derivative of f with respect to the third coordinate,

$$\partial_3 f(\mathbf{v}) = -\frac{v_3}{\sigma^2} f(\mathbf{v}).$$

Therefore,

$$W'(0) = -\frac{1}{\sigma^2}\cdot\int_{\mathbb{R}^3} f(\mathbf{v})\cdot A(\|\mathbf{v}\|)\cdot v_3^2\cdot d\mathbf{v}.$$

This integral is easily computed by using spherical coordinates, in terms of which $d\mathbf{v} = r^2 \sin\theta\, dr\, d\theta\, d\phi$,

$$W'(0) = \frac{1}{\sigma^2}\int_{r=0}^{\infty} A(r) e^{-\frac{r^2}{2\sigma^2}} r^4 \times \left[\int_{\theta=0}^{\pi} \sin\theta \cos^2\theta \left(\int_{\phi=0}^{2\pi} d\phi\right) d\theta\right] dr.$$

After performing the integrations on ϕ and θ, we get

$$W'(0) = -\frac{4\pi}{3\sigma^2}\int_{0}^{\infty} A(r) e^{-\frac{r^2}{2\sigma^2}} r^4 dr = -k_1,$$

where the number k_1 is clearly > 0. This is all we need to know, but just to have an idea of the magnitude of k_1, and of its dependence on σ, we plotted k_1 as a function of σ, observing that, except for values of σ at <2 or 3 Å, the linear relationship 5.25σ is a highly accurate approximation of $k_1(\sigma)$. (Take into account that this depends on the value of r_0, which we fixed at 1.5 Å.) Therefore, $W(z) \approx -k_1 z$, and separation of variables in Eq. 37 gives

$$\int_{z_0}^{z(t)} \frac{dz}{z} = -\frac{k_1}{b_1} t.$$

Hence $z(t) \approx z_0\, e^{-(k_1/b_1)t}$; therefore, the convergence of the trajectory is fast.

APPENDIX C: IMPLEMENTATION DETAILS

Besides the atomic-structure and map files, a few other pieces of input data are needed to run the code:

—A residue data file. This contains a list of residues along with codes specifying which variables in each residue are to be allowed to be free. (For the application to transition pathways or homology modeling, there is another column specifying the corresponding residue in the target structure.)

—The minimum RMS displacement between output conformations. A low value will produce many files and would be adequate to make a smooth movie of the trajectory. We used 1 Å for all of our examples, except in the rigid-domain part of EF-G, where 2 Å was used.

—The resolution of the EM map, in Å.

—The density cutoff level to remove the background of the map.

—The damp/drag ratio. This is the ratio between the scaling factors of the damping constants C^0_{kl} (Eq. 8) and of the friction constants b_{k} (Eq. 10) (see below). We used the default value of 10 for all of our examples.

The damp/drag ratio is the only user-specified parameter that affects the dynamics. A low value would in general allow more distortion of the structure, while a high value makes the structure more rigid; this might be needed in ''hard'' cases, but at the cost of a slower trajectory. The value of 10 used by us worked well in all our tests. The damping constants C^0_{kl} and the friction constants b_{k} were defined as

$$C^0_{\mathrm{kl}} = S_{\mathrm{damp}},$$

$$b_{\mathrm{k}} = S_{\mathrm{drag}} w^{\mathrm{at}}_{\mathrm{k}}/10,$$

where $w^{\mathrm{at}}_{\mathrm{k}}$ is the atomic weight of atom k, and S_{damp}, and S_{drag} are the scaling factors referred to above. The damp/drag ratio is then

$$\mathrm{damp/drag\ ratio} = S_{\mathrm{damp}}/S_{\mathrm{drag}}.$$

It is easy to see, from Eq. 12, that the dynamics depends only on this ratio; a change of the scaling factors that keeps the ratio constant will at most produce a change in the timescale, which is irrelevant. The same would happen if a scaling factor were used for the map force.

The code is freely available from the authors upon request. All molecular images were generated by the ICM software (22). We are grateful to Pablo Chacón for kindly providing suggestions to improve the manuscript.

This work was supported by National Institutes of Health grants No. R01 HL048908 (to M.Y.) and No. R01 GM071872 (to R.A.).

REFERENCES

1. Wriggers, W., R. K. Agrawal, D. L. Drew, J. A. McCammon, and J. Frank. 2000. Domain motions of EF-G bound to the 70S ribosome: insights from a hand-shaking between multi-resolution structures. *Biophys. J.* 79:1670–1678.
2. Wriggers, W., and S. Birmanns. 2001. Using *Situs* for flexible and rigid-body fitting of multi-resolution single-molecule data. *J. Struct. Biol.* 133:193–202.
3. Wriggers, W., and P. Chacón. 2001. Modeling tricks and fitting techniques for multi-resolution structures. *Structure.* 9:779–788.
4. Darst, S., N. Opalka, P. Chacón, A. Polyakov, C. Richter, G. Zhang, and W. Wriggers. 2002. Conformational flexibility of bacterial RNA polymerase. *Proc. Natl. Acad. Sci. USA.* 99:4296–4301.
5. Wriggers, W., P. Chacón, J. A. Kovacs, F. Tama, and S. Birmanns. 2004. Topology representing neural networks reconcile biomolecular shape, structure, and dynamics. *Neurocomputing.* 56:365–379.
6. Tama, F., W. Wriggers, and C. L. Brooks. 2002. Exploring global distortions of biological macromolecules and assemblies from low-resolution structural information and elastic network theory. *J. Mol. Biol.* 321:297–305.
7. Chacón, P., F. Tama, and W. Wriggers. 2003. Mega-Dalton biomolecular motion captured from electron microscopy reconstructions. *J. Mol. Biol.* 326:485–492.
8. Tama, F., O. Miyashita, and C. L. Brooks III. 2004a. Flexible multi-scale fitting of atomic structures into low-resolution electron density maps with elastic network normal mode analysis. *J. Mol. Biol.* 337:985–999.
9. Tama, F., O. Miyashita, and C. L. Brooks III. 2004b. Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM. *J. Struct. Biol.* 147:315–326.
10. Delarue, M., and P. Dumas. 2004. On the use of low-frequency normal modes to enforce collective movements in refining macromolecular structural models. *Proc. Natl. Acad. Sci. USA.* 101:6957–6962.
11. Suhre, K., J. Navaza, and Y.-H. Sanejouand. 2006. *NORMA*: a tool for flexible fitting of high-resolution protein structures into low-resolution electron-microscopy-derived density maps. *Acta Crystallogr. D Biol. Crystallogr.* 62:1098–1100.
12. Hinsen, K., N. Reuter, J. Navaza, D. L. Stokes, and J.-J. Lacapère. 2005. Normal mode-based fitting of atomic structure into electron density maps: application to sarcoplasmic reticulum Ca-ATPase. *Biophys. J.* 88:818–827.
13. Chen, J. Z., J. Fürst, M. S. Chapman, and N. Grigorieff. 2003. Low-resolution structure refinement in electron microscopy. *J. Struct. Biol.* 144:144–151.
14. Fabiola, F., and M. S. Chapman. 2005. Fitting of high-resolution structures into electron microscopy reconstruction images. *Structure.* 13:389–400.
15. Topf, M., and A. Sali. 2005. Combining electron microscopy and comparative protein structure modeling. *Curr. Opin. Struct. Biol.* 15: 578–585.
16. Topf, M., M. L. Baker, M. A. Marti-Renom, W. Chiu, and A. Sali. 2006. Refinement of protein structures by iterative comparative modeling and cryoEM density fitting. *J. Mol. Biol.* 357:1655–1668.
17. Velazquez-Muriel, J.-Á., M. Valle, A. Santamaría-Pang, I. A. Kakadiaris, and J.-M. Carazo. 2006. Flexible fitting in 3D-EM guided by the structural variability of protein superfamilies. *Structure.* 14:1115–1126.

18. Topf, M., K. Lasker, B. Webb, H. Wolfson, W. Chiu, and A. Sali. 2008. Protein structure fitting and refinement guided by Cryo-EM density. *Structure*. 16:295–307.

19. Jolley, C. C., S. A. Wells, P. Fromme, and M. F. Thorpe. 2008. Fitting low-resolution cryo-EM maps of proteins using constrained geometric simulations. *Biophys. J.* 94:1613–1621.

20. Wells, S., S. Menor, B. Hespenheide, and M. F. Thorpe. 2005. Constrained geometric simulation of diffusive motion in proteins. *Phys. Biol.* 2:S127–S136.

21. Jacobs, D. J., A. J. Rader, L. A. Kuhn, and M. F. Thorpe. 2001. Protein flexibility predictions using graph theory. *Proteins Struct. Funct. Bioinf.* 44:150–165.

22. Abagyan, R., M. Totrov, and D. Kuznetsov. 1994. ICM: a new method for structure modeling and design: applications to docking and structure prediction from the distorted native conformation. *J. Comput. Chem.* 15:488–506.

23. Mazur, A., and R. Abagyan. 1989. New methodology for computer-aided modeling of biomolecular structure and dynamics. 1. Non-cyclic structures. *J. Biomol. Struct. Dyn.* 6:815–832.

24. Abe, H., W. Braun, T. Noguti, and N. Gō. 1984. Rapid calculation of first and second derivatives of conformational energy with respect to dihedral angles for proteins. General recurrent equations. *Comput. Chem.* 8:239–247.

25. Xu, J. 2005. Rapid protein side-chain packing via tree decomposition. *In* Research in Computational Molecular Biology, Proceedings of the 9th Annual International Conference, RECOMB 2005, May 14–18, 2005. Vol. 3500 of Lecture Notes in Computer Science. S. Miyano, J. P. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. S. Waterman, editors. Springer-Verlag, Cambridge, MA.

26. Goldstein, H., C. P. Poole, C. P. Poole, Jr., and J. L. Safko. 2002. Classical Mechanics, 3rd Ed. Prentice Hall, Englewood Cliffs, NJ.

27. Morse, P. M., and H. Feshbach. 1953. Methods of Theoretical Physics. McGraw-Hill, New York.

28. Kabsch, W., H. G. Mannherz, D. Suck, E. F. Pai, and K. C. Holmes. 1990. Atomic structure of the actin:DNase I complex. *Nature*. 347:37–44.

29. Situs Modeling Package. 2006. http://situs.biomachina.org/.

30. Stern, D. F., P. Zheng, D. R. Beidler, and C. Zerillo. 1991. Spk1, a new kinase from *Saccharomyces cerevisiae*, phosphorylates proteins on serine, threonine, and tyrosine. *Mol. Cell. Biol.* 11:987–1001.

31. Chacón, P., and W. Wriggers. 2002. Multi-resolution contour-based fitting of macromolecular structures. *J. Mol. Biol.* 317:375–384.

32. Kovacs, J. A., and W. Wriggers. 2002. Fast rotational matching. *Acta Crystallogr. D Biol. Crystallogr.* 58:1282–1286.

33. Navaza, J., J. Lepault, F. A. Rey, C. Álvarez-Rúa, and J. Borge. 2002. On the fitting of model electron densities into EM reconstructions: a reciprocal-space formulation. *Acta Crystallogr. D Biol. Crystallogr.* 58:1820–1825.

34. Garzón, J. I., J. Kovacs, R. Abagyan, and P. Chacón. 2007. ADP_EM: fast exhaustive multi-resolution docking for high-throughput coverage. *Bioinformatics*. 23:427–433.

35. Toyoshima, C., M. Nakasako, H. Nomura, and H. Ogawa. 2000. Crystal structure of the calcium pump of sarcoplasmic reticulum at 2.6 Å resolution. *Nature*. 405:647–655.

36. Zhang, P., C. Toyoshima, K. Yonekura, N. M. Green, and D. L. Stokes. 1998. Structure of the calcium pump from sarcoplasmic reticulum at 8 Å resolution. *Nature*. 392:835–839.

37. Czworkowski, J., J. Wang, T. A. Steitz, and P. B. Moore. 1994. The crystal structure of elongation factor G complexed with GDP, at 2.7 Å resolution. *EMBO J.* 13:3661–3668.

38. Ævarsson, A., E. Brazhnikov, M. Garber, J. Zheltonosova, Y. Chirgadze, S. Al-Karadaghi, L. A. Svensson, and A. Liljas. 1994. Three-dimensional structure of the ribosomal translocase: elongation factor G from *Thermus thermophilus*. *EMBO J.* 13:3669–3677.

39. Laurberg, M., O. Kristensen, K. Martemyanov, A. T. Gudkov, I. Nagaev, D. Hughes, and A. Liljas. 2000. Structure of a mutant EF-G reveals domain III and possibly fusidic acid binding site. *J. Mol. Biol.* 303:593–603.

40. Valle, M., A. Zavialov, J. Sengupta, U. Rawat, M. Ehrenberg, and J. Frank. 2003. Locking and unlocking of ribosomal motions. *Cell*. 114: 123–134.

41. Müller, C. W., and G. E. Schulz. 1992. Structure of the complex between adenylate kinase from *Escherichia coli* and the inhibitor Ap$_5$A refined at 1.9 Å resolution. a model for a catalytic transition state. *J. Mol. Biol.* 224:159–177.

42. Kirillova, S., J. Cortés, A. Stefaniu, and T. Siméon. 2007. An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins. *Proteins Struct. Funct. Bioinf.* 70: 131–143.

43. Maragakis, P., and M. Karplus. 2005. Large amplitude conformational change in proteins explored with a plastic network model: adenylate kinase. *J. Mol. Biol.* 352:807–822.

44. Finnin, M. S., J. R. Donigian, and N. P. Pavletich. 2001. Structure of the histone deacetylase SIRT2. *Nat. Struct. Biol.* 8:621–625.

45. Zhao, K., R. Harshaw, X. Chai, and R. Marmorstein. 2004. Structural basis for nicotinamide cleavage and ADP-ribose transfer by NAD$^+$-dependent Sir2 histone/protein deacetylases. *Proc. Natl. Acad. Sci. USA*. 101:8563–8568.

46. Kufareva, I., L. Budagyan, E. Raush, M. Totrov, and R. Abagyan. 2007. PIER: protein interface recognition for structural proteomics. *Proteins Struct. Funct. Bioinf.* 67:400–417.