Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods

Ibrahim Al-Bluwi[1,2], Marc Vaisset[1,2],Thierry Siméon[1,2] and Juan Cortés[*1,2]

[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2]Univ de Toulouse, LAAS, F-31400 Toulouse, France

Email: Ibrahim Al-Bluwi - al.bluwi@gmail.com; Marc Vaisset - marc.vaisset@laas.fr; Thierry Siméon - nic@laas.fr; Juan Cortés*- juan.cortes@laas.fr;

*Corresponding author

Abstract

Background: Obtaining atomic-scale information about large-amplitude conformational transitions in proteins is a changeling problem for both experimental and computational methods. Such information is, however, important for understanding the mechanisms of interaction of many proteins. This paper presents a computationally efficient approach, combining methods originating from robotics and computational biophysics, to model protein conformational transitions. The capacity of normal mode analysis to predict directions of collective large-amplitude motions is exploited to bias the conformational exploration performed by a motion planning algorithm. A coarse-grained elastic network model built on short fragments of three residues is proposed to reduce the dimension of the problem, and for the rapid computation of normal modes. The accurate reconstruction of the all-atom model from the coarse-grained one is achieved using closed-form inverse kinematics.

Results: Tests on a set of ten proteins demonstrate the capacity of the method to model conformational transitions of proteins within a few hours of computing time on a single processor. These results also show that the computing time scales linearly with the protein size, independently of the protein topology. Further experiments on adenylate kinase show that main features of the transition between the open and closed conformations of this protein are well captured in the computed path.

Conclusions: The proposed method enables the simulation of large-amplitude conformational transitions in proteins using very few computational resources. The resulting paths are a first approximation that can directly provide important information on the molecular mechanisms involved in the conformational transition. This approximation can be subsequently refined and analyzed using state-of-the-art energy models and molecular modeling methods.

Keywords: Protein conformational transitions; elastic network models; normal mode analysis; motion planning algorithms; inverse kinematics.

Background

Studying conformational transitions in proteins is important for understanding their biological functions, since such motions are generally related to their capacity to interact with other molecules. However, capturing this type of dynamic information at the atomic scale is difficult using experimental techniques. Modeling protein conformational transitions with conventional computational methods is also challenging since, in many cases, these transitions are rare, slow events. Standard molecular dynamics (MD) simulations with current computational resources cannot be applied in practice to model large-amplitude (slow time-scale) conformational transitions. Such simulations require variants of MD methods that enhance sampling of rare events or that bias the exploration in a given direction (e.g. [1–5]), or, alternatively, to have access to outstanding computational power [6].

Modeling conformational transitions in proteins has motivated the development of specific methods, computationally more efficient than MD simulations. Many of these methods (e.g. [7–9]) are based on the deformation of an initial path toward the minimum energy path between the two given conformations. Consequently, the performance of these methods is strongly conditioned by the quality of the initial path, which can be difficult to obtain. Methods to model conformational transitions have also been developed based on robot motion planning algorithms [10–13]. For the sake of efficiency, these methods usually deal with simplified molecular models. Therefore, they are mainly aimed at providing qualitative information about the conformational transition.

The main difficulty to be faced by all types of computational methods to model protein conformation transitions is the high dimensionality of the space to be explored. In this regard, normal mode analysis (NMA) [14] is an interesting tool for dimensionality reduction. Indeed, a reduced number of low-frequency normal modes are a good indicator of the direction of large-amplitude conformational changes [15–18]. Several recent works exploit this property of NMA to enhance the performance of conformational exploration

methods (e.g. [19, 20]).

This paper presents a variant of the method introduced in [19]. The method combines the rapidly-exploring random tree (RRT) algorithm [21] and NMA to model protein conformational transitions. The main novelty presented here concerns the introduction of a multi-scale model for the protein. A coarse-grained model that considers a single particle per tripeptide is used to define an elastic network on which NMA is performed. Motion directions provided by the normal modes are then applied to the all-atom model for a finer conformational exploration. The introduction of this multi-scale model has important outcomes. Using the coarse-grained model, the number of normal modes is significantly reduced, which greatly decreases the time required to compute them. Besides, moving between the coarse-grained and the all-atom models can be achieved accurately and efficiently using methods from robot kinematics [22], without the need of artifacts such as the RTB approach (rotations-translations of blocks) [23].

Next section presents the overall method, and explains each of the elementary components: elastic network normal mode analysis, tripeptide-based multi-scale protein modeling, and motion-planning-based conformational exploration. Then, several types of results that validate the approach and show its good computational performance are presented for a set of proteins with different sizes and topologies. A more detailed analysis of results is presented for adenylate kinase. Finally, together with the conclusions, we discuss possible directions for future work.

Methods

This section presents a new method to model protein conformation transitions. It builds on the combination of three components. One of these components is NMA performed on a coarse-grained elastic network model of the protein, which enables very fast computation of normal modes. Indeed, a single particle of the elastic network is considered for each group of three consecutive amino-acid residues (i.e. one particle per tripeptide). The all-atom model, which is used to accept or reject sampled states during the conformational exploration, is accurately reconstructed from the coarse-grained one using closed-form inverse kinematics. The overall algorithm to compute the conformational transition path proceeds iteratively. At each iteration, the RRT algorithm is applied to explore linear combinations of normal modes computed from intermediate conformations along the path. All these elementary components of the method are further explained below.

Elastic Networks and Normal Mode Analysis

Every molecule has a set of natural vibration modes, called *normal modes*, that depends on its structure. Each mode corresponds to a motion pattern, in which all atoms of the molecule move with the same frequency and in phase, i.e. all passing through the equilibrium and maximum points at the same time. It has been shown that low-frequency normal modes correspond to collective atomic motions (or domain motions), whereas high-frequency normal modes correspond to local fluctuations [16, 24].

Normal modes can be calculated by diagonalizing the Hessian matrix of the potential energy of the molecule. For reducing the computational cost of this operation, several works propose to use simplified potentials and coarse-grained models. An extensively used simplified potential is based on the elastic network model (ENM) [25], which represents the molecule as a set of particles connected by virtual springs. All the protein atoms can be considered as particles in this model. However, a coarse-grained representation is usually applied by considering C_α atoms only, i.e. a single particle per amino-acid residue [16,17]. Moreover, particles are connected by virtual springs only if the distance between them is less than a user-defined cut-off distance d_{cut}. The potential energy function of such an elastic network takes the following form:

$$E = \sum_{d_{ij}^0 < d_{cut}} \frac{C}{2}(d_{ij} - d_{ij}^0)^2$$

where d_{ij} is the distance between particle i and particle j, d_{ij}^0 is the distance between the two particles at the equilibrium state and C is the elastic constant. This type of simplified potential has been used in many works and for very different applications [26–29].

Here, we investigate a further simplification of the ENM. Instead of using C_α atoms, we build the ENM using a simplified representation based on tripeptides. Figure 1 illustrates the approach. Note that coarse-grained NMA approaches considering more than one residue per particle have been proposed [23, 30, 31]. However, these approaches, which are mainly devised to analyze conformational changes of very large systems made of protein assemblies, consider rigid-body motions of groups of residues. In contrast, the approach presented here preserves full flexibility of the protein, which leads to a more accurate modeling of conformational transitions.

It has been shown that using a simplified ENM does not necessarily lead to a loss of accuracy in the prediction of motion directions [17, 23]. However, it certainly leads to a reduction in computing time. Note that using tripeptides instead of C_α atoms reduces the size of the Hessian matrix by a factor of 3, which significantly reduces the computing time required for diagonalization. This issue is discussed in more details in the results section.

The anisotropic network model (ANM) approach, as described in [24, 32], is adopted in this work to construct the Hessian matrix from the positions of the particles of the tripeptide-based model. Each 3×3 sub-matrix corresponding to the interaction between two particles is computed as follows:

$$H_{ij} = -\frac{C}{d_{ij}^2} \begin{bmatrix} (x_j - x_i)(x_j - x_i) & (x_j - x_i)(y_j - y_i) & (x_j - x_i)(z_j - z_i) \\ (y_j - y_i)(x_j - x_i) & (y_j - y_i)(y_j - y_i) & (y_j - y_i)(z_j - z_i) \\ (z_j - z_i)(x_j - x_i) & (z_j - z_i)(y_j - y_i) & (z_j - z_i)(z_j - z_i) \end{bmatrix}$$

$$H_{ii} = -\sum_{j|j\neq i} H_{ij}$$

If the distance between particles i and j is more than the cut-off distance d_{cut}, then the whole 3×3 matrix is replaced by zeros. The Hessian matrix is then diagonalized to compute the eigenvalues and eigenvectors. Each eigenvalue and eigenvector pair corresponds to one normal mode, where the eigenvalue defines the mode frequency and the eigenvector defines the motion direction for each particle in the elastic network.

Multi-Scale Model

Tripeptide-based Model

The multi-scale modeling approach applied in this work is based on a decomposition of the protein chain into fragments of three amino acid residues, which we refer to as *tripeptides*. The reason for choosing such a subdivision is that the backbone of a tripeptide involves 6 degrees of freedom (three pairs of angles ϕ, ψ)[1], and thus, an analogy can be made with a $6R$ mechanism like a robotic manipulator [22]. Reference frames attached to the N atom in the backbone of the first residue and to the C atom in the last residue define respectively the *base-frame* and the *end-frame* of the $6R$ mechanism. Since tripeptides are linked through rigid peptide bonds, the location of the end-frame of tripeptide i can be determined from the base-frame of tripeptide $i + 1$ by a constant transformation. Given the location of the base-frame and the end-frame, the conformation of a tripeptide backbone can be determined by *inverse kinematics*. Consequently, the conformation of the whole protein backbone can be determined from the pose of a single reference frame attached to each tripeptide[2]. In the following, we will refer to these reference frames as (oriented) *particles*. They are the particles in the coarse-grained ENM. Further explanations on this tripeptide-based modeling approach can be found in [33], where the model is used for the implementation of move classes within Monte Carlo methods.

[1] Bond lengths and bond angles, as well as peptide bond torsions are considered to have constant values.

[2] The affirmation is true for all the protein backbone except two short fragments at the N-terminal and C-terminal ends of the chain, which require a particular treatment.

Reconstructing the All-Atom Model

The interest of the decomposition of the protein into tripeptides explained above is that closed-form inverse kinematics (IK) can be applied to reconstruct the protein backbone conformation from the coordinates of the particles. The IK solver applied in this work has been adapted from the method developed by Renaud [34]. This solver is based on algebraic elimination theory, and develops an ad-hoc resultant formulation inspired by the work of Lie and Liang [35]. Starting from a system of equations representing the IK problem, the elimination procedure leads to an 8-by-8 quadratic polynomial matrix in one variable. The problem can then be treated as a generalized eigenvalue problem, as proposed in [36], for which efficient and robust methods such as the Schur factorization can be applied. Note however that our approach is not dependent on this solver, so that other IK methods (e.g. [36, 37]) could be applied.

In general, the IK problem for a 6R serial kinematic chain has a finite number of solutions (up to 16 in the most general case). All the solutions correspond to geometrically valid conformations of the tripeptide backbone with fixed ends defined by the pose of the particles. However, when the goal is to simulate continuous motions, the closest conformation to the previous one (i.e. the one before a perturbation applied to the particles) has to be selected in order to avoid jumps in the conformational space. If none of the solutions remains within a distance threshold that depends on the perturbation step-size, the IK problem is considered to have no solution.

The explanations above concern only the reconstruction of the all-atom model of the protein backbone from the coarse-grained tripeptide-based model. Side-chains are treated separately, using a simple method based on energy minimization as explained below.

Path Finding Algorithm

The proposed method works by iteratively creating short portions of the conformational transition path between two given conformations of a protein, which we will refer to as q_{init} and q_{goal}. The steps of the algorithm are summarized in Algorithm 1. At each iteration, the normal modes of a root conformation q_{root} are computed (q_{root} for the first iteration is q_{init}). These normal modes are then used to bias a short RRT exploration, which is run until the protein moves a predefined distance toward the target conformation q_{goal}. Further details on the conformational exploration performed by the RRT algorithm are given below. The closest node in the tree q_{close} to q_{goal} is then identified, and the path between q_{root} and q_{close} is extracted and saved. All the conformations in this path are guaranteed to have a collision-free backbone[3], which generally

[3] C_β atoms are considered to be part of the backbone.

Algorithm 1: COMPUTE_PATHWAY

```
input  : Initial conformation q_init, final conformation q_goal and minimum distance to target d_target
output : The transition pathway p
begin
    q_root ← q_init;
    while RMSD(q_root, q_goal) > d_target do
        a ← COMPUTE_NORMALMODES(q_root);
        t ← BUILD_RRT(a, q_root, q_goal);
        q_close ← CLOSESTTOTARGET(t, q_goal);
        q_root ← MINIMIZE(q_close);
        p ← CONCATENATE(p, q_root);
end
```

Algorithm 2: BUILD_RRT

```
input  : Initial conformation q_root, final conformation q_goal
output : The tree t
begin
    t ← INITTREE(q_root);
    while not STOPCONDITION(t, q_goal) do
        q_rand ← SAMPLE(t);
        q_near ← BESTNEIGHBOR(t, q_rand);
        q_new ← EXPANDTREE(q_near, q_rand);
        if ISVALID(q_new) then
            ADDNEWNODE(t, q_new);
            ADDNEWEDGE(t, q_near, q_new);
end
```

implies getting acceptable energy values after a short minimization to rearrange side-chain conformations. Such an energy minimization procedure is performed on q_{close}, which will be the root conformation in the next iteration. The algorithm keeps iterating until a predefined distance d_{target} from q_{goal} is reached. The resulting path is defined by the sequence of minimized conformations q_{close} at each iteration. If a finer grained path is required, other intermediate conformation can be extracted from the sub-paths computed at each iteration. These conformations may require energy minimization to rearrange side-chains, as it is done for q_{close}.

Implementation Details

The RRT algorithm iteratively applied in Algorithm 1 performs the same steps as the basic RRT [21]. The steps are sketched in Algorithm 2. At each iteration, a conformation q_{rand} is randomly sampled. Note that

q_{rand} is not required to be a feasible conformation. Then, the tree is searched for a conformation q_{near}, which is the closest conformation to q_{rand}. A new conformation, q_{new}, is generated by moving a predefined short distance from q_{near} towards q_{rand}. The new conformation is added to the tree if it does not violate feasibility constraints, which in the present work are limited to geometric constrains related to no atom overlapping and no bond breaking. The difference with respect to the basic RRT algorithm concerns the implementation of the methods for sampling conformations, searching the nearest neighbor, and expanding the tree, which are specific to the particular settings: the use of the multi-scale model of the protein, and the application of NMA to bias the exploration. The particularities of these three methods are explained next.

Sampling Random Conformations

The idea is to generate a random sample q_{rand} that allows the RRT to explore the conformational space using information given by the normal modes. This operation is performed on the coarse-grained model, thus using the set of particles. Hence, q_{rand} is not an all-atom conformation, but an array of particle positions. These positions are generated by moving the particles from q_{root} in the directions given by a linear combination of normal modes with randomly sampled weights. More precisely:

- A sequence of $3n$ random weights w_j are sampled in the range [-1, 1], where n is the number of particles, being $3n$ the number of normal modes[4].

- The new positions of the n particles are computed by a linear combination of all the randomly weighted modes as follows:

$$q_{rand} = q_{root} + \sum^{3n} f * w_j * a_j$$

  where a_j refers to each normal mode, and f is an amplification factor used to push the sampled conformation away from q_{root} (this factor is the same for all the normal modes). Note that since the normal modes are not normalized, low frequency modes have larger norm. Thus, they contribute more significantly in the sum.

Finding Nearest Neighbors

Nearest neighbor search is also performed using the coarse-grained model. Indeed, the computed distance is the root mean squared deviation (RMSD) of the particle positions. An additional bias is used in our

[4]Actually, the number of normal modes is $3n - 6$, since 6 degrees of freedom correspond to rigid-body motions of the whole set of particles.

implementation to pull the exploration towards the target conformation. The biased distance is computed as follows:

$$d(q, q_{rand}) = \text{RMSD}(q, q_{rand}) \frac{\text{RMSD}(q, q_{goal})}{\text{RMSD}(q_{init}, q_{goal})}.$$

In this work, we have implemented a simple brute-force algorithm to find q_{near}. However, more sophisticated nearest neighbor search algorithms based on space partitioning techniques (e.g. [38]) could be used to reduce the number of performed distance computations.

Generating New Conformations

In order to generate q_{new}, all particle positions in q_{near} are linearly interpolated towards q_{rand} with a predefined distance k. Given these new particle positions, the all-atom model corresponding to q_{new} is generated using IK. We apply an iterative process that solves IK for every tripeptide t_i using the new positions of particles p_i and p_{i+1}. If no IK solution is found for a tripeptide or if the solution found involves atom collisions, the pose of particle p_{i+1} is slightly perturbed for a new trial. Note that, in addition to the particle position, a small perturbation is also applied to the orientation, since the problem can be due to restraints caused by the current orientations of the particles. This process is repeated until a collision-free IK solution is found or a maximum number of trials is reached. If this process fails to find a collision-free IK solution for any tripeptide, failure is reported and the RRT algorithm goes back to the random sampling step.

After generating IK solutions for all the tripeptides, the only remaining parts of the protein backbone to be addressed are the two terminal fragments. The pose of these fragments is adjusted such that they are in accordance with the new poses of the first and last particles respectively. Random perturbations can be applied to the two end fragments in order to remove possible collisions.

The generated conformation q_{new} is guaranteed to satisfy hard geometric constraints since, as mentioned before, every generated tripeptide conformation is checked for collisions. However, in order to speed-up computations, side-chains are excluded in this test (only C_β atoms are considered). This is because side-chains are known to be very flexible, and resolving possible collisions along the paths can be done in a post-processing stage. Hence, any side-chain collision is assumed to be resolved during the minimization step at the end of each short RRT execution, as mentioned above.

Results and Discussion

This section discusses several experiments aimed to validate the proposed method and to evaluate its performance. First, the question concerning the accuracy of the tripeptide-based elastic network model is addressed. Then, results are presented on conformational transitions computed for a set of ten proteins with different sizes and topologies. Finally, further results on adenylate kinase are presented and compared to available data on the transition between the open and closed forms of this protein.

Validating the Coarse-Grained ENM

Previous works (e.g. [16,17]) have shown that simple ENMs built using C_α atoms perform as well as ENMs built using the all-atom model when studying the dynamic properties of proteins with NMA. Here, we compare the performance of the proposed tripeptide-based model with the C_α-based model for predicting directions of conformational transitions. A set of seven proteins listed in Table 1 was used for this comparison. These proteins were also used in related work [17] for the validation of the C_α-based ENM.

For evaluating the capability of normal modes to predict directions of conformations transitions, we use the notion of *overlap* as proposed in related work [17]. The overlap I_j between a normal mode j and an experimentally observed conformational change between two conformations (open and closed) q^o and q^c is defined as a measure of similarity between the conformational change and the direction given by the normal mode j. It can be computed as follows:

$$I_j = \frac{\left| \sum^{3n} a_{ij} \Delta q_i \right|}{\left[\sum^{3n} a_{ij}^2 \sum^{3n} \Delta q_i^2 \right]^{1/2}}$$

where $\Delta q_i = q_i^o - q_i^c$ measures the difference between the particle coordinates in conformations q^o and q^c, a_{ij} corresponds to the i^{th} coordinate of the normal mode j, and n is the number of particles. A value of 1 for the overlap means that the direction given by the normal mode matches exactly the conformational change, whereas a value around 0.2 or less means that the normal mode is unable to provide any meaningful prediction.

Before conducting the comparative analysis, we need to determine an optimal cutoff distance for the tripeptide-based ENM. A good cutoff distance should create an elastic network that correctly captures the topology of the protein. For C_α-based models, 8 Å is generally used, since this cutoff distance has been empirically shown to provide the best results in most cases. It can be intuitively inferred that the

same cutoff distance may not be the optimal choice in our case, because distances between particles of the tripeptide-based model are larger than distances between C_α atoms. To determine the optimal value, we have measured and compared overlap values for the seven proteins with cutoff distances between 8 and 34 Å. Figure 2 shows the average overlap value achieved for each cutoff distance over the seven proteins. The overlap value considered for each protein is the best one found among the overlap values of all the normal modes. As can be clearly seen in the figure, the highest averages are for cutoff distances of 15, 16 and 17 Å. This is coherent with the optimal distance of 8 Å suggested for C_α-based models because, although tripeptides involve three consecutive C_α atoms, they usually adopt conformations that are not fully extended. This means that the optimal cutoff distance for the tripeptide-based model is expected to be less than three times the optimal cutoff used for the C_α-based model. The tripeptide-based ENMs for four of the proteins in Table 1, using a cutoff distance of 16 Å, are represented in Figure 3. The figure shows that the main typological features of the proteins appear in the coarse-grained model.

Table 2 shows overlap values using a cutoff distance of 16 Å , and compares them to the values presented in [17] for the C_α-based ENM using a cutoff distance of 8 Å. In the table, columns labeled "Open" correspond to the case of moving from the open to the closed conformation and columns labeled "Closed" are for the opposite case. It is clear that both ENMs provide comparable overlap values, which means that our simplified ENM is also able to capture the topological information necessary for computing normal modes that correctly predict motion directions. Note that the overlap values can even be better if the best cutoff distance for each protein is used instead of always using 16 Å.

Importantly, such a similar performance in terms of overlap is obtained with a significant reduction of the computational cost. Since the computational complexity of the Hessian matrix diagonalization is $\mathcal{O}(n^3)$, the reduction of n by a factor 3 provides a theoretical gain of more than one order of magnitude. We have confirmed this theoretical gain with some experiments that show that the time required to compute the normal modes with our coarse-gained model ranges from 0.05 seconds to 0.9 seconds, while using the C_α model may require up to several minutes (detailed results are not presented here).

Finding Conformational Transitions

Experimental Setup

We have applied the proposed method to compute conformational transition pathways for the ten proteins listed in Table 3, and represented in Figure 4. For each protein, at least two experimental structures corresponding to different conformations are available in the Protein Data Bank (PDB) [39]. The difference

between these conformations involves large-amplitude domain motions. The ten proteins are varied in size and topology, as well as in the type of domain motions they undergo. This variability presents a challenge for the method, and makes the achieved results indicative of its performance and its scalability.

As mentioned in the previous section, each iteration of the method performs a short RRT exploration. In the current implementation, each RRT exploration runs until the protein has moved 0.3 Å C_α-RMSD towards the goal. This distance is gradually reduced to 0.15 Å as the distance to the goal becomes smaller. The reason is that the speed of convergence tends to decrease when approaching the target conformation, and recomputing normal modes more frequently provides better results in this situation. The exploration is stopped after a certain number of iterations (4000 in our case) if the distance stopping condition is not satisfied first. This additional stopping condition is introduced to prevent too long runs of RRT when it is unable to move the required distance towards the goal.

Once the RRT exploration stops, the closest conformation to the goal is identified and submitted to an energy minimization procedure aimed at generating better side-chain conformations. We have used in our experiments the AMBER software package [40] for energy minimization.

Results

Table 4 summarizes the results achieved by the proposed method for the set of ten proteins. In this table, C_α-RMSD$_{end}$ is the distance between the goal conformation and the closest conformation found by our method. The table also shows the total computing time and the partial time required by RRT. Time$_{total}$ includes Time$_{RRT}$ plus the time needed for computing the normal modes and running minimizations at each iteration. Finally, the number of iterations indicated in this table refers to the number of times normal modes have been computed. In all of the simulations, the RRT exploration takes more than 90% of the total time spent by the method. Note that simulations were run on a single core of an AMD Opteron 148 processor at 2.6 GHz.

Our method was able to model the conformational transition in all cases, reaching conformations very close to the given goal conformations. Figure 5 shows superimposed structures[5] of open and closed forms of the proteins (q_{init} and q_{goal}), and of the closed form and the last conformation of the computed transition path (q_{goal} and q_{final}). The distances between the final and goal conformations are below 2 Å (measured using C_α-RMSD) for all the tested proteins with the exception of DDT and GroEL. Note that 2 Å RMSD corresponds to the current resolution of accurate experimental methods for protein structure determination.

[5]Structure superimpositions and images have been done using PyMOL [41].

Even for the two proteins presenting worst results, DDT and GroEL, the superimpositions of the final and goal conformations shown in Figure 5 display their high similarity. Note that the method could have reached closer conformations to the goal, however, the strategy in our simulations was to stop when the distance to the goal reached a very slow convergence rate.

We have analyzed the relationship between the size of the protein and the computing time required by our method to model the conformational transition. Since the lengths of the transition paths are different for the different proteins, we have measured the time required to move 1Å along these paths. Results presented in Table 5 and Figure 6 show a linear scalability, which is an interesting property. Note that the topology of the protein seems to have no or little influence on the performance of the method. This is an important advantage with respect to the method presented in [19], which showed some difficulties when dealing with relative motions of domains connected through several linkers due to the internal-coordinate representation used to model proteins.

Finally, Table 6 shows the percentage of the time spent by our method performing some of the most time-consuming steps of the RRT exploration. Values are provided for nearest neighbor search (NN), collision checking (CC), inverse kinematics (IK) and random sampling (RS). An interesting observation in this table is that nearest neighbor search consumes around 60% of the computing time. This is mainly due to the brute-force nearest neighbor algorithm used in our implementation. As mentioned before, more sophisticated nearest neighbor algorithms can be used to overcome this performance bottleneck. The computational performance could also be improved by using simplified distance metrics that save computing time while preserving the quality of the exploration (e.g. [42, 43]).

A Closer Look at Adenylate Kinase

Adenylate kinase (ADK) [44] is a widely studied signal transduction protein. Its structure is divided into three main domains known as: LID, CORE and NMPbind. Several works (e.g. [45,46]) suggest that the LID and NMPbind domains undergo clear conformational changes, whereas the CORE domain remains almost unchanged. It has also been suggested that the transition between open and closed conformations of the protein goes through a two-step process where the NMPbind domain moves less clearly than the LID domain at the beginning, and then moves at a faster pace as the transition approaches its end [46].

Figure 7 shows the open and closed conformations of ADK (corresponding to PDB IDs 4AKE and 1AKE, respectively) along with several intermediate conformations generated by our method. As expected, the LID and NMPbind domains change significantly compared to the CORE domain. Figure 8 shows the displacement

of the residues along the conformational transition, where darker regions represent larger displacements. Regions around residues 20-60 and 130-160, which approximately correspond to the NMPbind and LID domains respectively, are clearly highlighted. It is also clear in the plot that residues of the NMPbind domain start moving with more significance near the end of the conformational transition, whereas residues in the LID domain start at an earlier stage, which reflects the two-step nature of the transition discussed earlier. These results show that the path generated by our method is in agreement with previous results, including those presented in our previous work [19].

We have also compared intermediate conformations in the computed transition path of the ADK to a small number of other experimentally solved structures of this protein. These structures correspond to homolog proteins or mutants with very high sequence identity, and some of them are known to be intermediate structures between open and closed forms of the protein. Interestingly, four of these structures are very close to conformations along the transition path. Table 7 shows the distance between each of these structures and the closest conformation in the transition path. The table also shows the position of the this conformation in the path. More precisely, the table shows the corresponding iteration number and the percentage of the path length. 2RH5 (A) is very close to the conformation generated by the first iteration, whereas 1E4Y (A) is close to the conformation generated by iteration 27 (near the closed structure). 1DVR (A) is also very close to a conformation toward the beginning of the path (near the open structure), whereas 2RH5 (B) is a slightly less open structure. These results are comparable to those provided by previous studies [12, 47], which further validates the proposed method.

Conclusions

This paper has presented an efficient method for computing large-amplitude motions in proteins. The proposed method makes use of both the ability of normal modes to locally predict motion directions and the efficiency of the RRT algorithm to explore large spaces. Using normal modes alone would require performing a large number of iterations, and RRT alone would waste time in exploring irrelevant parts of the conformational space. Hence, combining the two methods allows overcoming the drawbacks of each one separately. The proposed approach also relies on the tripeptide-based representation of the protein, which reduces the number of computed modes and provides an accurate method for switching between the coarse-grained model and the all-atom model.

Performed experiments show that computing normal modes of a protein using the coarse-grained tripeptide-based model instead of the C_α atoms to define an ENM does not lead to a degradation in the

ability to predict motion directions, while the computing time is significantly reduced. Results also show that the proposed method is able to model large-amplitude conformational transitions in proteins of different sizes and topologies, and that computing time scales linearly with the number of residues. Using an unoptimized implementation, computing time ranges from a few hours in small proteins to a few days in large ones. This time could be significantly reduced by the implementation of more sophisticated methods to perform the most costly operations within the RRT algorithm.

An interesting extension that could be implemented to improve the computational performance of our method is the use of a bi-directional RRT [21], which constructs two trees rooted at the initial and goal conformations respectively. In additions, a parallelized version of RRT could also provide a significant performance gain [48]. Finally, using T-RRT [49] instead of RRT could also be an interesting direction for future work. In this case, the aim will not be to improve the performance in terms of computing time, but in terms if path quality. Indeed, paths computed with T-RRT should follow more accurately the valleys of the conformational energy landscape [50].

In this work, we have demonstrated the capacity of the proposed method to compute transition paths between two given conformations of a protein. However, the approach could also be applied to a more challenging problem: the prediction of other (meta-)stable states reachable from a given protein conformation. This more challenging problem would require some extensions, mainly in the definition of scoring functions to identify interesting intermediate and meta-stable states during the conformational exploration.

List of abbreviations

MD: molecular dynamics; NMA: normal mode analysis; RRT: rapidly-exploring random tree; RTB: rotations-translations of blocks; ENM: elastic network model; ANM: anisotropic network model; IK: inverse kinematics; RMSD: root mean squared deviation; PDB: Protein Data Bank.

Author's contributions

TS and JC designed this research and supervised the work. IA implemented the method and carried out experiment. MV participated in the software design and implementation. IA and JC wrote the manuscript. All authors have read and approved the manuscript.

Acknowledgements

This work has been partially supported by the French National Research Agency (ANR) under project ProtiCAD.

References

1. Voter AF: **A method for accelerating the molecular dynamics simulation of infrequent events**. *J. Chem. Phys.* 1997, **106**(11):4665–4677.
2. Izrailev S, Stepaniants S, Isralewitz B, Kosztin D, Lu H, Molnar F, Wriggers W, Schulten K: **Steered molecular dynamics**. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*, Springer-Verlag 1998:39–65.
3. Sørensen RR, Voter AF: **Temperature-accelerated dynamics for simulation of infrequent events**. *J. Comput. Phys.* 2000, **112**:9599–9606.
4. Laio A, Parrinello M: **Escaping free-energy minima**. *Proc Natl. Acad. Sci. U.S.A.* 2002, **99**(20):12562–12566.
5. Hamelberg D, Morgan J, McCammon JA: **Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules**. *J. Chem. Phys.* 2004, **120**:11919–11929.
6. Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, Bank JA, Jumper JM, Salmon JK, Shan Y, Wriggers W: **Atomic-level characterization of the structural dynamics of proteins**. *Science* 2010, **330**(6002):341–346.
7. Mills G, Jónsson H: **Quantum and thermal effects in H2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems**. *Phys. Rev. Lett.* 1994, **72**:1124–1127.
8. E W, Ren W, Vanden-Eijnden E: **String method for the study of rare events**. *Phys. Rev. B.* 2002, **66**:052301.
9. Bolhuis PG, Chandler D, Dellago C, Geissler PL: **Transition path sampling and the calculation of rate constants**. *Annu. Rev. Phys. Chem.* 2002, **53**:291–318.
10. Cortés J, Siméon T, Ruiz de Angulo V, Guieysse D, Remaud-Siméon M, Tran V: **A path planning approach for computing large-amplitude motions of flexible molecules**. *Bioinformatics* 2005, **21**(suppl 1):i116–i125.
11. Raveh B, Enosh A, Schueler-Furman O, Halperin D: **Rapid sampling of molecular motions with prior information constraints**. *PLoS Comput. Biol.* 2009, **5**(2):e1000295.
12. Haspel N, Moll M, Baker M, Chiu W, Kavraki LE: **Tracing conformational changes in proteins**. *BMC Struct. Biol.* 2010, **10**(Suppl 1):S1.
13. Al-Bluwi I, Siméon T, Cortés J: **Motion planning algorithms for molecular simulations: a survey**. *Comput. Sci. Rev.* 2012, **6**(4):125–143.
14. Cui Q, Bahar I: *Normal mode analysis: theory and applications to biological and chemical systems*. Chapman and Hall/CRC mathematical and computational biology series, Chapman & Hall/CRC 2006.
15. Brooks B, Karplus M: **Normal modes for specific motions of macromolecules: application to the hinge-bending mode of lysozyme**. *Proc Natl. Acad. Sci. U.S.A.* 1985, **82**(15):4995–4999.
16. Hinsen K: **Analysis of domain motions by approximate normal mode calculations**. *Proteins* 1998, **33**(3):417–429.
17. Tama F, Sanejouand YH: **Conformational change of proteins arising from normal mode calculations**. *Prot. Eng.* 2001, **14**:1–6.
18. Alexandrov V, Lehnert U, Echols N, Milburn D, Engelman D, Gerstein M: **Normal modes for predicting protein motions: a comprehensive database assessment and associated Web tool**. *Prot. Sci.* 2005, **14**(3):633–643.
19. Kirillova S, Cortés J, Stefaniu A, Siméon T: **An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins**. *Proteins* 2008, **70**:131–143.
20. Cossins BP, Hosseini A, Guallar V: **Exploration of protein conformational change with PELE and meta-dynamics**. *J. Chem. Theory Comput.* 2012, **8**:959–965.

21. LaValle SM, Kuffner JJ: **Rapidly-exploring random trees : progress and prospects**. In *Algorithmic and Computational Robotics: New Directions*. Edited by Donald B, Lynch K, Rus D, Boston: A.K. Peters 2001:293–308.
22. Siciliano B, Khatib O: *Springer Handbook of Robotics*. Springer 2008.
23. Tama F, Gadea FX, Marques O, Sanejouand YH: **Building-block approach for determining low-frequency normal modes of macromolecules**. *Proteins* 2000, **41**:1–7.
24. Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I: **Anisotropy of fluctuation dynamics of proteins with an elastic network model**. *Biophys. J.* 2001, **80**:505–515.
25. Tirion MM: **Large amplitude elastic motions in proteins from a single-parameter, atomic analysis**. *Phys. Rev. Lett.* 1996, **77**(9):1905–1908.
26. Kim MK, Jernigan RL, Chirikjian GS: **Efficient generation of feasible pathways for protein conformational transitions**. *Biophys. J.* 2002, **83**(3):1620–1630.
27. Tama F, Miyashita O, Brooks III CL: **Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM**. *J. Struc. Biol.* 2004, **147**(3):315–326.
28. Cavasotto CN, Kovacs JA, Abagyan RA: **Representing receptor flexibility in ligand docking through relevant normal modes**. *J. Am. Chem. Soc.* 2005, **127**(26):9632–9640.
29. Mouawad L, Perahia D: **Motions in hemoglobin studied by normal mode analysis and energy minimization: evidence for the existence of tertiary T-like, quaternary R-like intermediate structures**. *J. Mol. Biol.* 1996, **258**(2):393–410.
30. Schuyler AD, Chirikjian GS: **Efficient determination of low-frequency normal modes of large protein structures by cluster-NMA**. *J. Mol. Graph. Model.* 2005, **24**:46–58.
31. Demerdash ONA, Mitchell JC: **Density-cluster NMA: a new protein decomposition technique for coarse-grained normal mode analysis**. *Proteins* 2012, **80**(7):1766–1779.
32. Eyal E, Yang LW, Bahar I: **Anisotropic network model: systematic evaluation and a new web interface**. *Bioinformatics* 2006, **22**(21):2619–2627.
33. Cortés J, Al-Bluwi I: **A robotics approach to enhance conformational sampling of proteins**. *Proc. IDETC/CIE* 2012.
34. Renaud M: **A simplified inverse kinematic model calculation method for all 6R type manipulators**. In *Current Advances in Mechanical Design and Production VII*. Edited by Hassan MF, Megahed SM, New York: Pergamon 2000:57–66.
35. Lee HY, Liang CG: **A new vector theory for the analysis of spatial mechanisms**. *Mech. Mach. Theory* 1988, **23**(3):209–217.
36. Manocha D, Canny JF: **Efficient inverse kinematics for general 6R manipulators**. *IEEE Trans. Robot. Autom.* 1994, **10**(5):648–657.
37. Coutsias EA, Seok C, Jacobson MP, Dill KA: **A kinematic view of loop closure**. *J. Comput. Chem.* 2004, **25**(4):510–528.
38. Atramentov A, LaValle SM: **Efficient nearest neighbor searching for motion planning**. *Proc. IEEE Int. Conf. Robot. Autom.* 2002, :632–637.
39. **Research Collaboratory for Structural Bioinformatics PDB: http://www.rcsb.org/pdb/**.
40. Case DA, Darden TA, Cheatham III TE, Simmerling CL, Wang J, Duke RE, Luo R, Merz KM, Pearlman DA, Crowley M, et al.: *AMBER 9*. San Francisco: University of California 2006.
41. **The PyMOL Molecular Graphics System, Version 1.5, Schrödinger, LLC.**
42. Plaku E, Stamati H, Clementi C, Kavraki L: **Fast and reliable analysis of molecular motion using proximity relations and dimensionality reduction**. *Proteins* 2007, **67**(4):897–907.
43. Shehu A, Olson B: **Guiding the search for native-like protein conformations with an ab-initio tree-based exploration**. *Int. J. Robot. Res.* 2010, **29**(8):1106–1127.
44. Müller CW, Schulz GE: **Structure of the complex between adenylate kinase from Escherichia coli and the inhibitor Ap5A refined at 1.9 Å resolution. A model for a catalytic transition state**. *J. Mol. Biol.* 1992, **224**:159–177.

45. Müller CW, Schlauderer GJ, Reinstein J, Schulz GE: **Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding**. *Structure* 1996, **4**(2):147–156.

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

47. Feng Y, Yang L, Kloczkowski A, Jernigan RL: **The energy profiles of atomic conformational transition intermediates of adenylate kinase**. *Proteins* 2009, **77**(3):551–558.

48. Devaurs D, Siméon T, Cortés J: **Parallelizing RRT on distributed-memory architectures**. *Proc. IEEE Int. Conf. Robot. Autom.* 2011, :2261–2266.

49. Jaillet L, Cortés J, Siméon T: **Sampling-based path planning on configuration-space costmaps**. *IEEE Trans. Robot.* 2010, **26**(4):635–646.

50. Jaillet L, Corcho FJ, Pérez JJ, Cortés J: **Randomized tree construction algorithm to explore energy landscapes**. *J. Comput. Chem.* 2011, **32**(16):3464–3474.

Figures

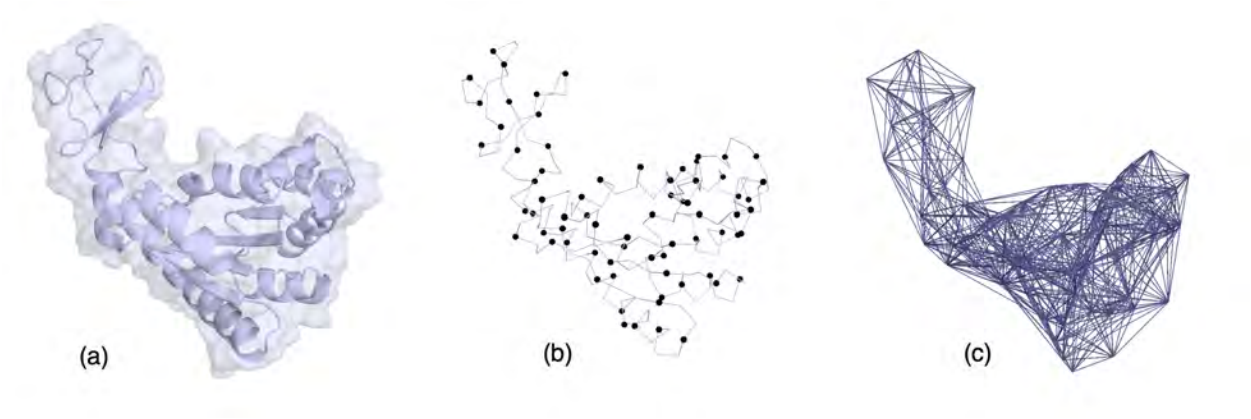


Figure 1 - Illustration of the different models on the ADK protein:

(a) Representation of the all-atom model, (b) the particles of the coarse-grained tripeptide-based model, (c) representation of the elastic network model.

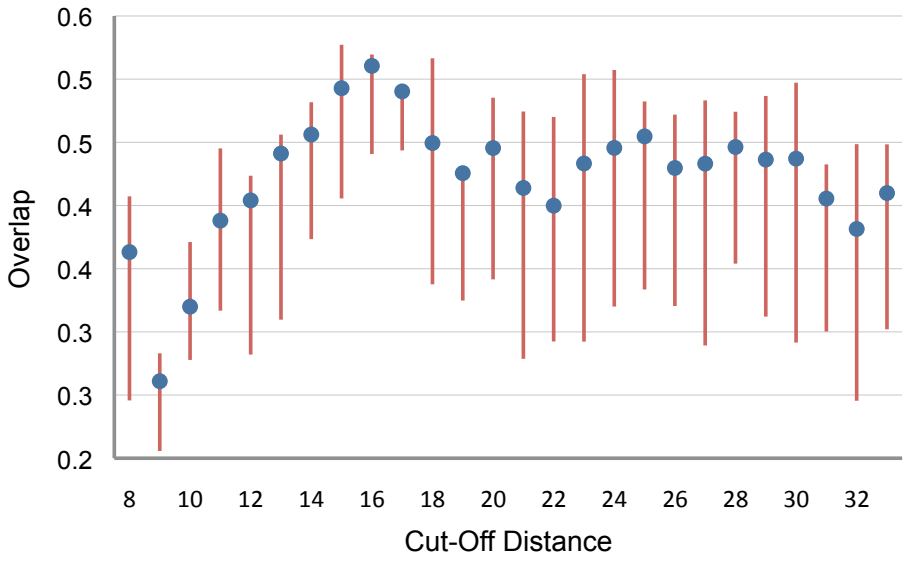


Figure 2 - Average overlap over the seven proteins of Table 1

Lines are drawn between the 25^{th} and the 75^{th} percentiles of the overlap values. Average overlap values are indicated with dots.

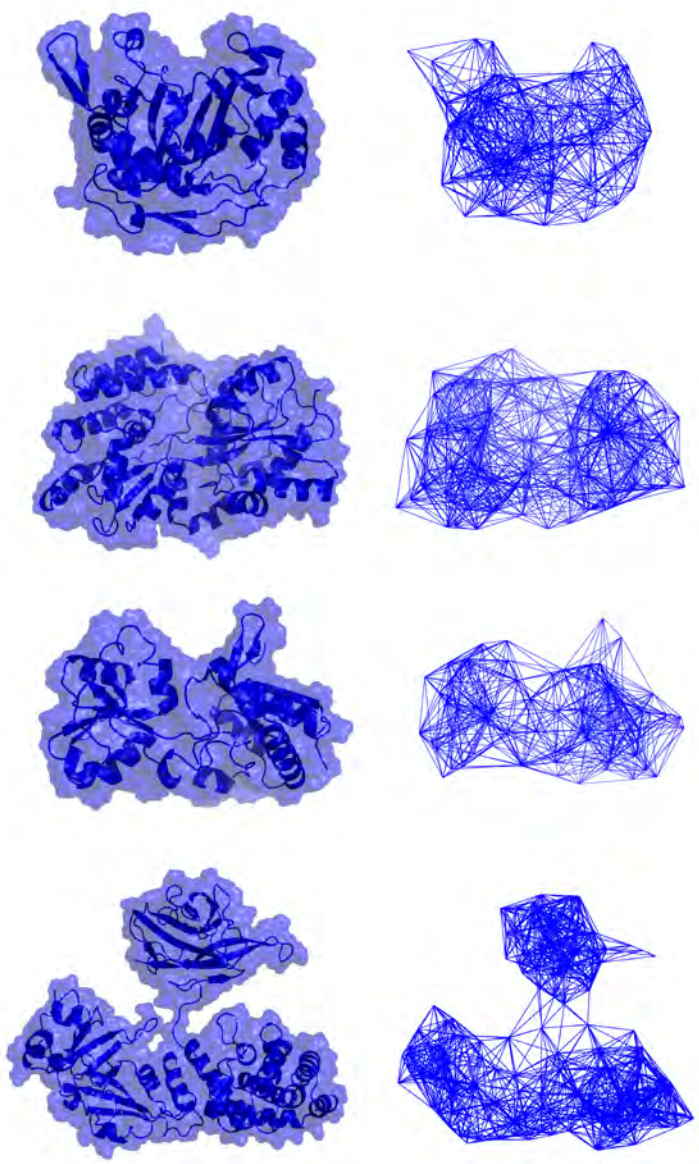


Figure 3 - Tripeptide-based elastic network models

Representation of the all-atom models and the tripeptide-based ENMs for four different proteins.

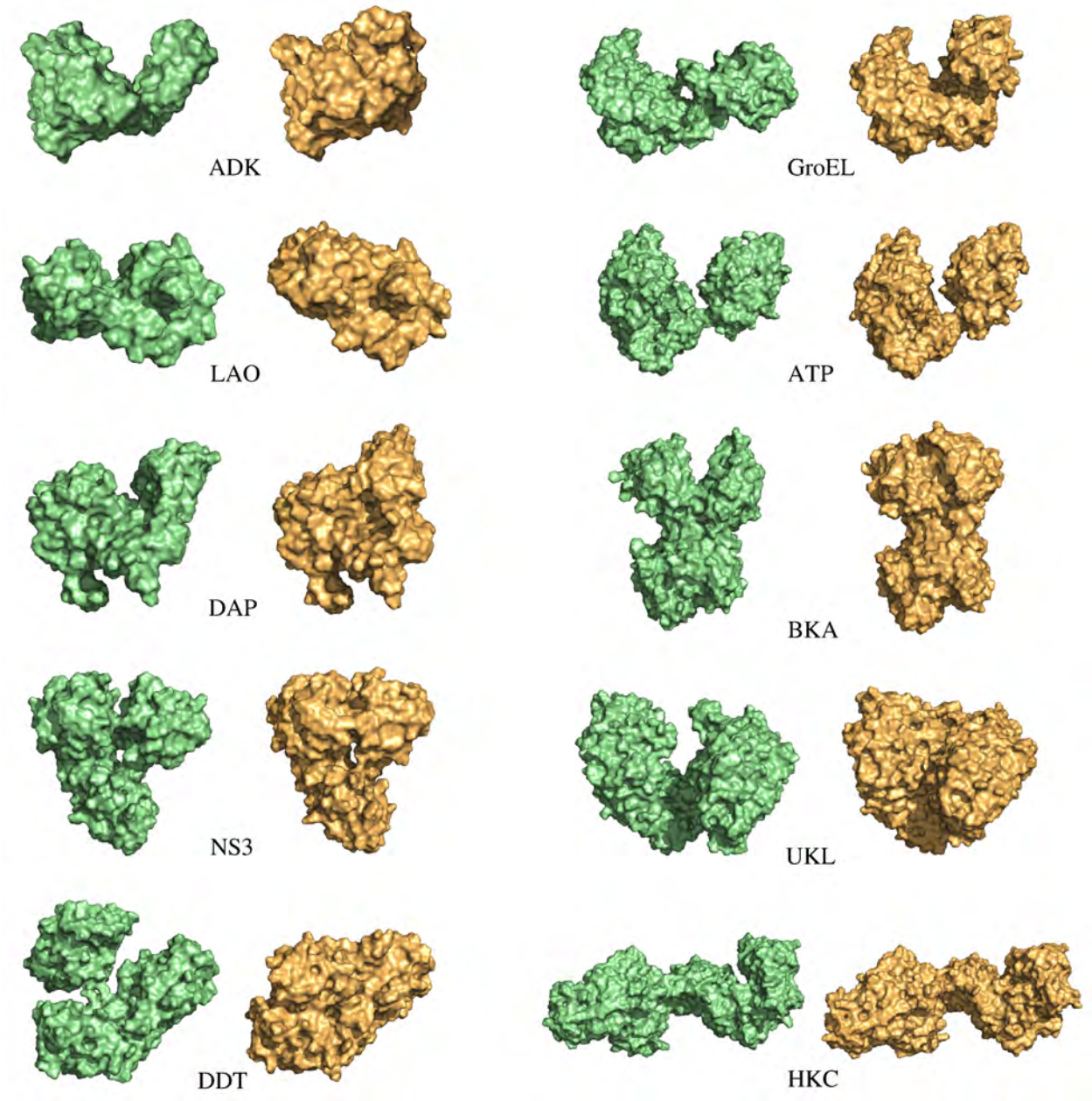


Figure 4 - The ten proteins used in the experiments

Representation of open and closed forms of these proteins available in the PDB (IDs are provided in Table 3).

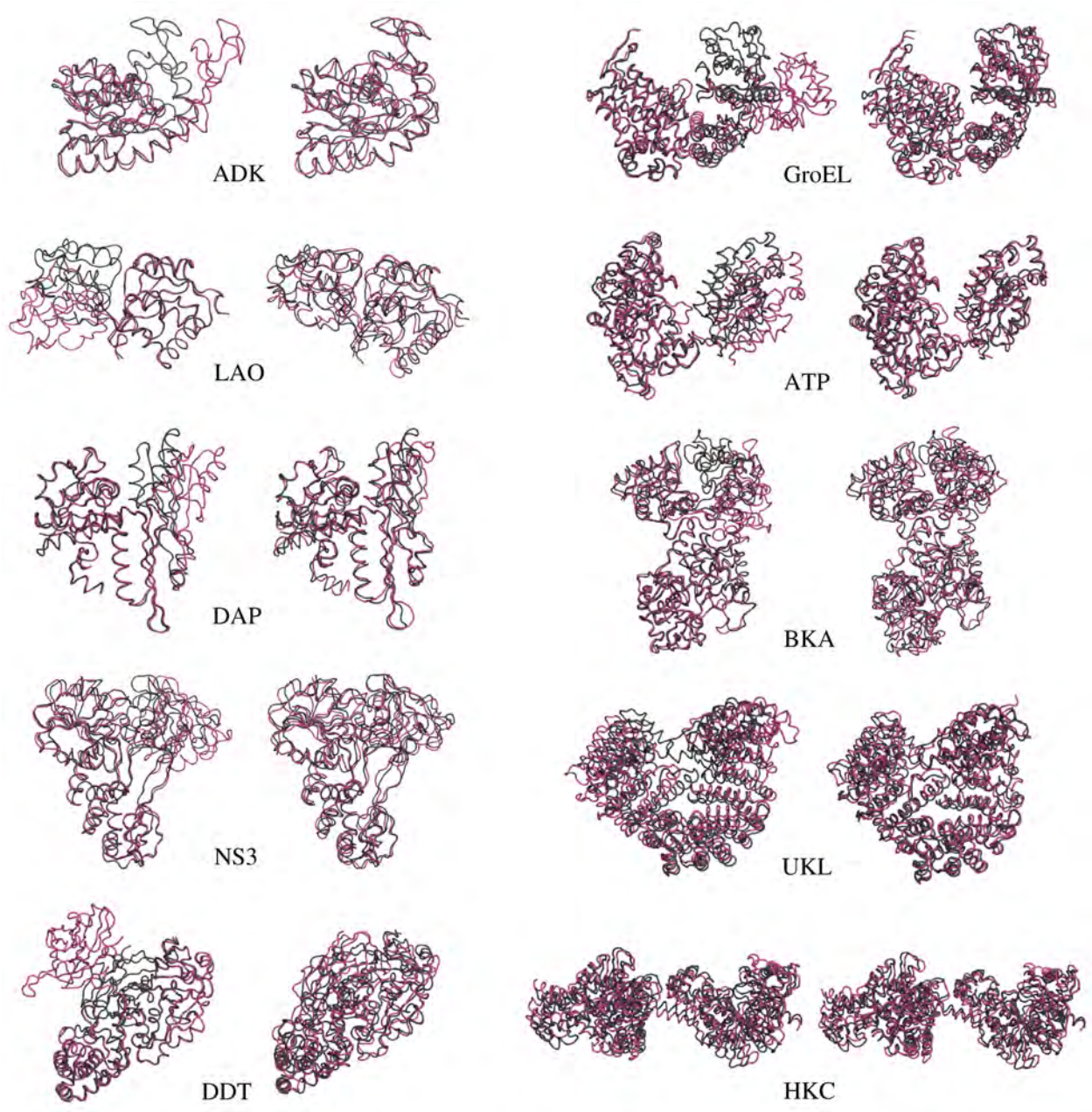


Figure 5 - Superimposed structures and final conformations of the computed transition path

For each protein, the left image shows the open form (in red) and the closed form (in black), and the right image shows the closed form (in black) and the final conformations of the computed path (in red).

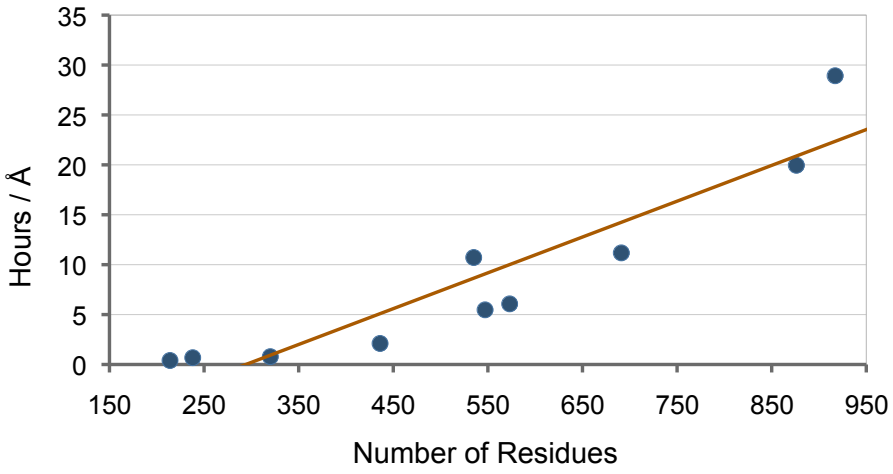


Figure 6 - Plot of the results in Table 5

The plot shows a linear relationship between the size of the protein and the time required to compute the conformational transition path.

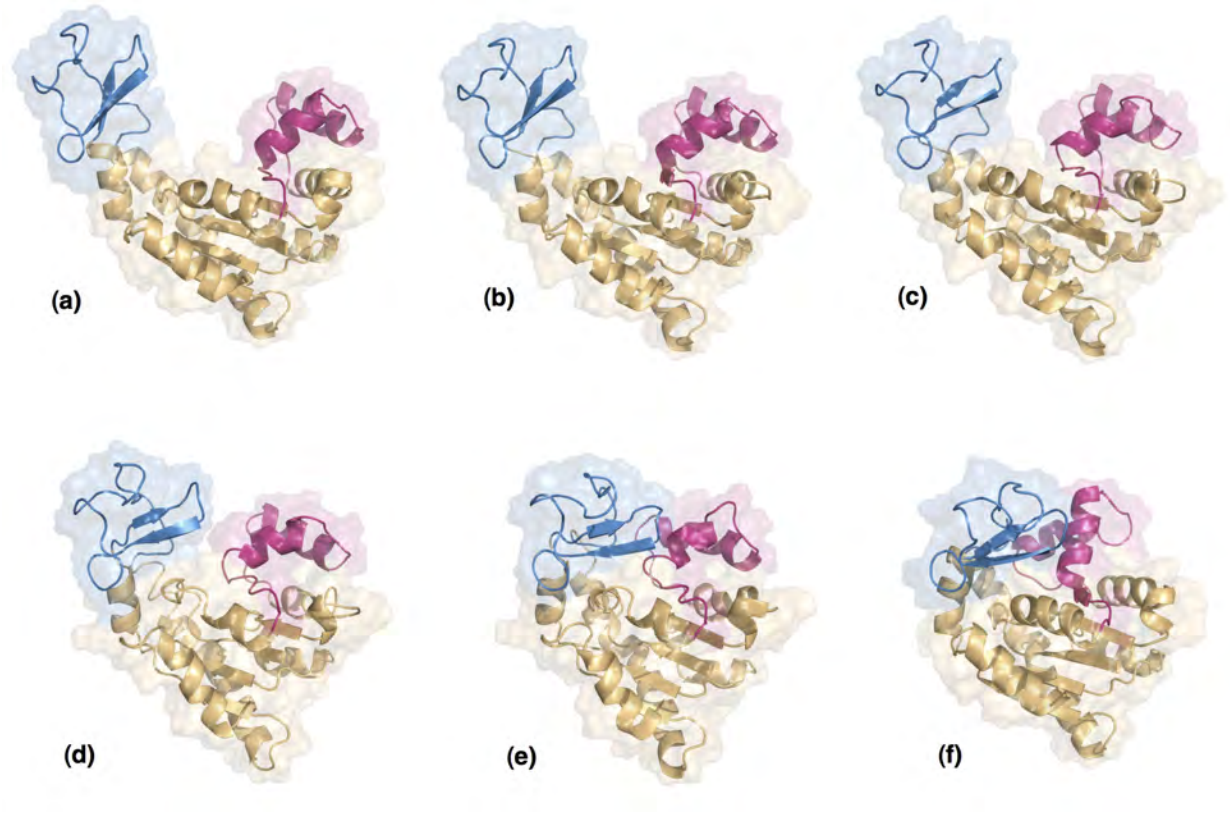


Figure 7 - Different conformations of ADK along the studied conformational transition

The LID domain is shown in blue and the NMPbind domain is shown in red. Images (a) and (f) represent the start and goal conformations respectively. Images (b) to (e) show intermediate conformations generated by our method.

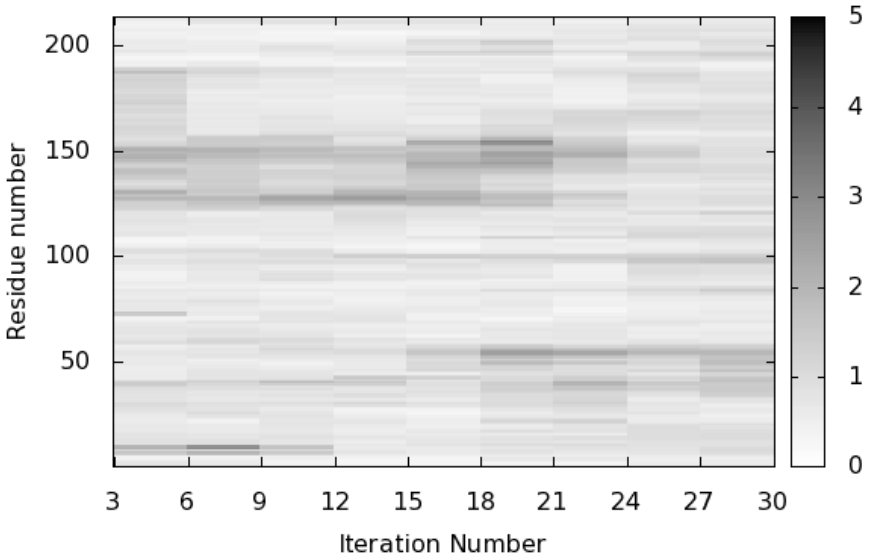


Figure 8 - Displacement of the residues along the conformational transition

The plot shows, using a gray-scale, the displacement of each residue at each iteration relative to the previous iterations. Darker regions represent larger displacements.

Tables

Table 1 - Proteins used in the overlap experiments

Protein	Residues	PDB_{open}	PDB_{closed}
Che Y Protein	128	3chy	1chn
LAO binding Protein	238	2lao	1laf
Triglyceride Lipase	256	3tgl	4tgl
Thymidulate Synthase	264	3tms	2tsc
Maltodextrine Binding Protein	370	1omp	1anf
Enolase	436	3enl	7enl
Diphtheria Toxin	523	1ddt	1mdt

Table 2 - Comparison between overlap values for C_α-based ENMs and tripeptide-based ENMs

Protein	C_α Overlap		Tripeptides Overlap	
	Open	Close	Open	Close
Che Y Protein	0.32	0.34	0.52	0.34
LAO binding Protein	0.84	0.40	0.53	0.52
Triglyceride Lipase	0.30	0.17	0.26	0.35
Thymidulate Synthase	0.56	0.40	0.49	0.29
Maltodextrine Binding Protein	0.86	0.77	0.90	0.84
Enolase	0.33	0.30	0.40	0.30
Diphtheria Toxin	0.58	0.37	0.48	0.30

Table 3 - Proteins used in the experiments

Protein	Residues	PDB ID_{init}	PDB ID_{goal}	C_α RMSD
ADK	214	4ake	1ake	6.51
LAO	238	2lao	1laf	3.73
DAP	320	1dap	3dap	3.78
NS3	436	3kqk	3kql	2.75
DDT	535	1ddt	1mdt	10.96
GroEL	547	1aon	1oel	10.49
ATP	573	1m8p	1i2d	3.78
BKA	691	1cb6	1bka	4.75
UKL	876	1ukl	1qgk	6.17
HKC	917	1hkc	1hkb	3.00

Table 4 - Performance of the method on ten proteins (cf. Table 3)

Protein	C_α-$RMSD_{end}$	Iterations	$Time_{RRT}$	$Time_{total}$
ADK	1.56	31	1.82	2.00
LAO	1.32	20	1.52	1.65
DAP	1.31	16	1.78	1.92
NS3	1.29	14	2.82	3.00
DDT	2.88	272	81.54	86.4
GroEL	2.79	142	40.21	42.17
ATP	1.45	30	13.46	14.16
BKA	1.96	74	29.56	31.09
UKL	1.99	80	80.61	82.62
HKC	1.64	38	37.91	39.63

Table 5 - Relationship between the size of the protein and the computing time

Protein	Residues	Time (hours)
ADK	214	0.4
LAO	238	0.68
DAP	320	0.79
NS3	436	2.11
DDT	535	10.72
GroEL	547	5.84
ATP	573	6.74
BKA	691	11.17
UKL	876	19.96
HKC	917	28.93

Table 6 - Percentage of the time spent performing the main RRT operations

Protein	NN	CC	IK	RS
ADK	57.2%	14.1%	15.0%	6.3%
LAO	51.3%	20.9%	17.0%	5.4%
DAP	50.5%	20.6%	11.0%	12.3%
NS3	67.9%	13.4%	6.6%	8.9%
DDT	64.3%	17.1%	6.9%	9.0%
GroEL	60.4%	17.6%	8.9%	9.8%
ATP	57.3%	20.9%	6.8%	11.9%
BKA	55.1%	16.8%	6.1%	19.3%
UKL	62.9%	15.5%	4.1%	15.5%
HKC	68.9%	5.8%	3.3%	18.2%
Average	59.58%	16.27%	8.57%	11.66%

Table 7 - Known intermediate structures and their distances to the closest conformation in the computed transition path.

PDB ID	RMSD	Iteration	Path percent
1DVR (A)	1.48	2	9%
2RH5 (A)	1.80	1	4%
2RH5 (B)	1.91	3	15%
1E4Y (A)	2.20	27	94%

Additional Files

adk.mov — Movie of a conformational transition path for AKD

This movie has been generated with QuickTime and saved in the native file format *.mov*. It can be viewed using other video players.

OTHER VIDEOS ....