Coarse-grained elastic networks, normal mode analysis and robotics-inspired methods for modeling protein conformational transitions

Ibrahim Al Bluwi, Marc Vaisset, Thierry Simeon, Juan Cortés

Coarse-grained elastic networks, normal mode analysis and robotics-inspired methods for modeling protein conformational transitions

Ibrahim Al-Bluwi, Marc Vaisset, Thierry Siméon, Juan Cortés
CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ de Toulouse, LAAS, F-31400 Toulouse, France
Email: {ialbluwi, marc, nic, jcortes}@laas.fr

***Abstract*—This paper presents a method, inspired by robot motion planning algorithms, to model conformational transitions in proteins. The capacity of normal mode analysis to predict directions of collective large-amplitude motions is exploited to bias the conformational exploration. A coarse-grained elastic network model built on short fragments of three residues is proposed 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 show the capacity of the method to model conformational transitions of proteins within a few hours of computing time on a single processor. Tests on a set of ten proteins demonstrate 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.**

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

I. Introduction

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 methods. 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], [2], [3], [4], [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], [8], [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], [11], [12], [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], [16], [17], [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 node 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 briefly discusses normal mode analysis performed on elastic network models of proteins. Then, Section III presents the multi-scale model used in this work, which is based on a decomposition of the protein into tripeptides and on the application of inverse kinematics to reconstruct the all-atom model. The proposed method, which combines the aforementioned concepts within an RRT-based algorithm to model protein conformational transitions, is described in Section IV. Results presented in Section V compare the accuracy of tripeptide-based NMA with C_α-based NMA, and demonstrate the good performance of the overall approach to model protein conformational transitions. Results are presented for a set of proteins with different sizes and topologies.

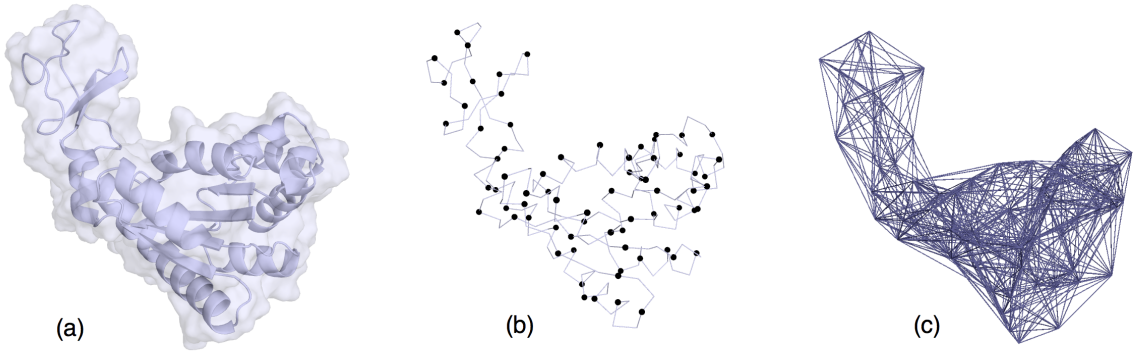

Fig. 1. Illustration of the different models on the ADK protein: (a) Representation of the all-atom model, (b) the nodes of the coarse-grained tripeptide-based model, (c) representation of the elastic network model.

II. 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]. The ENM represents the molecule as a set of nodes connected by virtual springs. Moreover, nodes are connected only if the distance between them is less than a user-defined cutoff distance. All the protein atoms can be considered as nodes in this model. However, a coarse-grained representation is usually applied by considering C_α atoms only, i.e. a single node per amino-acid residue [16], [17].

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 (see Section III). Figure 1 illustrates the approach. Note that coarse-grained NMA approaches considering more than one residue per node have been proposed [23], [26], [27]. 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 [23], [17]. 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 Section V-A.

Finally, we have adopted in this work the Anisotropic Network Model (ANM) approach as described in [24], [28] to construct the Hessian matrix from the positions of the nodes of the tripeptides-based model.

III. Multi-Scale Model

A. 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]. A reference frame attached to the N atom in the backbone of the first residue defines the *base-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*. These particles are the nodes of the coarse-grained ENM. Further explanations on this tripeptide-based modeling approach can be found in [29], 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.

B. 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 [30]. This solver is based on algebraic elimination theory, and develops an ad-hoc resultant formulation inspired by the work of Lie and Liang [31]. 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 [32], 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. [32], [33]) could be applied.

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 in next section.

IV. Method

A. Overall 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 implies having acceptable energy values. In order to rearrange side-chains, a quick minimization step 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 final trajectory is defined by the sequence of minimized conformations q_{close} at each iteration. If a finer grained trajectory is required, then the extracted paths at each iteration can be used, which may require further minimization.

B. Implementation Details

The RRT algorithm iteratively applied in Algorithm 1 performs the same steps as the standard RRT [21]. The steps are sketched in Algorithm 2. However, the implementation of the methods for sampling, searching the nearest neighbor, and expanding the tree 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.

[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
        m ← COMPUTE_NORMALMODES(q_root);
        t ← BUILD_RRT(m, 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
```

1) 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 $3n-6$ random weights w_i are sampled in the range of [-1, 1], where n is the number of particles ($3n-6$ in the number of normal modes).
- The array of the new positions of the n particles is created using a linear combination of all the weighted modes. More precisely, the position of a particle i is computed

as follows:

$$p_i^{new} = p_i^{root} + \sum_{j=6}^{j<3n} w_j * f * nm_j,$$

where p_i^{root} is the position of the particle i in q_{root}, nm_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).

2) Finding Nearest Neighbors: Nearest neighbor search is also performed on 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 implementation to pull the exploration towards the target conformation. The biased distance is computed as follows:

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

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. [34]) could be used to reduce the number of performed distance computations.

3) 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 could 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 in above.

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 I
PROTEINS USED IN THE OVERLAP EXPERIMENTS.

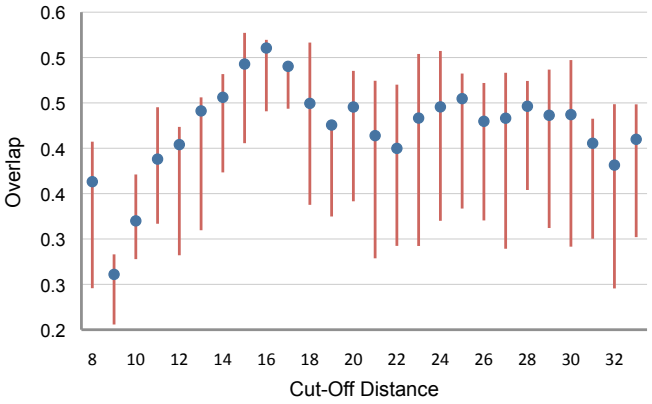

Fig. 2. Average overlap over the seven proteins of Table I. Lines are drawn between the 25^{th} and the 75^{th} percentiles of the overlap values. Average overlap values are indicated with dots.

V. Experiments and Results

A. Validating the Coarse-Grained ENM

Previous works such as [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 coarse-grained model with the C_α-based model for predicting directions of conformational transitions. For this, we use the notion of *overlap* as proposed in related works (see [17] for details on the computation of the overlap).

We have measured overlap values for the seven proteins listed in Table I. These proteins were also used in [17] for the validation of the C_α-based ENM. Overlap values provided in [17] correspond to a cutoff distance of 8 Å, which is a standard value used for this type of simplified models. A good cutoff distance should create an elastic network that correctly captures the topology of the protein. We have measured and compared overlap values for the seven proteins with cutoff distances between 8 and 34 Å to find the optimal one for the coarse-grained tripeptide-based ENM. 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 cutoffs 15, 16 and 17. 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

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 II
COMPARISON BETWEEN OVERLAP VALUES FOR ENMS BUILT USING THE COARSE-GRAINED TRIPEPTIDE-BASED MODEL AND ENMS BUILT USING C_α ATOMS AS PRESENTED IN [17].

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.

Table II shows overlap values using a cutoff distance of 16 Å and compare them to the values presented in [17] for the C_α-based ENM. 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).

B. Finding Conformational Transitions

1) Experimental Setup: We have applied the proposed method to compute conformational transition pathways for the ten proteins listed in Table III. For each protein, at least two experimental structures corresponding to different conformations are available in the Protein Data Bank (PDB) [35]. The difference between these conformations involves large-amplitude domain motions. The ten proteins are varied in size and in the type of domain motions they undergo. This variability presents a challenge for the method, which makes the achieved results indicative of its performance and its scalability.

As mentioned in Section IV-A, each iteration of the method performs a short RRT exploration. In our experiments, 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 generating new valid conformations is harder in the vicinity of the closed conformation since the compactness of the conformation reduces the free space. 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 minimized. We have used in our experiments the AMBER software package [36] for the minimization. The Eigen software library[4] has been applied for the computation of the normal modes, i.e. for the diagonalization of the Hessian matrix.

2) Results: Table IV summarizes the results achieved by our method for the ten proteins. In this table, 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

[4]http://eigen.tuxfamily.org/.

Protein	Residues	PDB ID_{init}	PDB ID_{goal}	RMSD_{init}
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 III
PROTEINS USED IN THE EXPERIMENTS. IN THIS TABLE, RMSD IS THE ROOT MEAN SQUARED DEVIATION COMPUTED USING THE C_α ATOMS.

Protein	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 IV
PERFORMANCE OF THE METHOD ON TEN PROTEINS (CF. TABLE III).

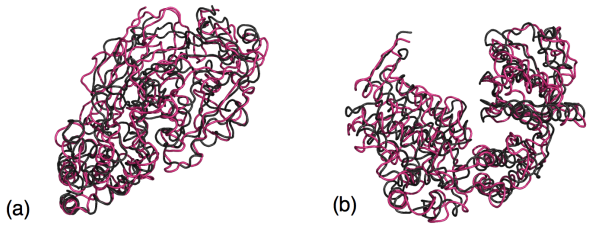

Fig. 3. Superimposed final (red) and goal (black) conformations for proteins: (a) DDT and (b) GroEL.

Protein	Residues	Speed (hours / Å)
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 V
RELATIONSHIP BETWEEN THE SIZE OF THE PROTEIN AND THE TIME REQUIRED TO COMPUTE PATHS OF 1Å C_α-RMSD.

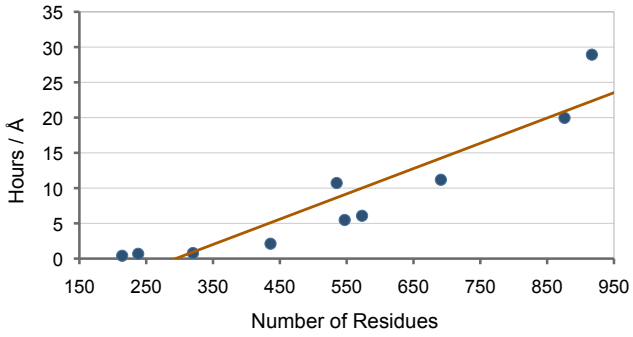

Fig. 4. Plot of the results in Table V, showing a linear relationship between the size of the protein and the computing time.

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 VI
PERCENTAGE OF THE TIME SPENT PERFORMING THE MAIN RRT OPERATIONS: NN (NEAREST NEIGHBOR SEARCH), CC (COLLISION CHECKING), IK (INVERSE KINEMATICS) AND RN (RANDOM SAMPLING).

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. 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. Even for the two proteins presenting worst results, DDT and GroEL, the superimpositions[5] of the final and goal conformations shown in Figure 3 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 V and Figure 4 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 VI shows the percentage of the time spent by our method performing some of the most time-consuming steps of the RRT exploration. 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. [37], [38]).

3) A Closer Look at Adenylate Kinase: Adenylate Kinase (ADK) [39] 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. [40], [41]) 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 [41].

[5]The superimposition of the conformations has been performed using the software package PyMol (http://www.pymol.org/)

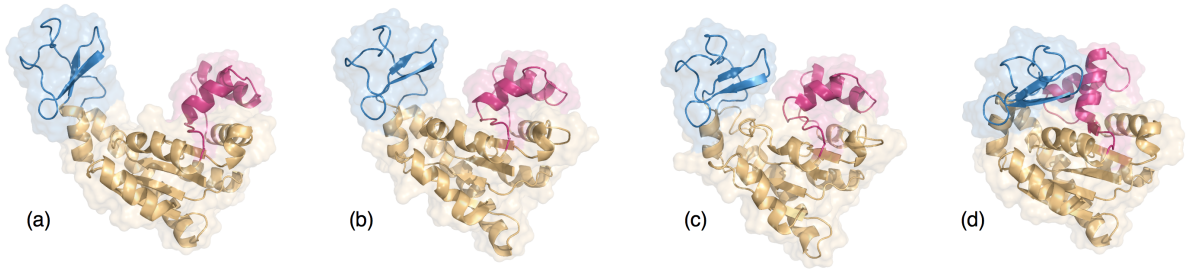

Fig. 5. Different conformations of the ADK protein along the studied conformational transition. The LID domain is shown in blue and the NMPbind domain is shown in red. Conformations (a) and (d) are the start and goal conformations respectively. (b) and (c) are intermediate conformations generated by our method.

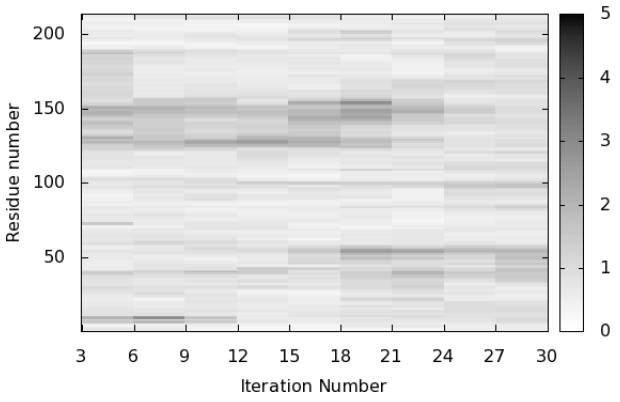

Fig. 6. Displacement of the residues at each iteration relative to the previous iterations. Darker regions represent larger displacements.

Figure 5 shows the open and closed conformations of ADK (corresponding to PDB IDs 4AKE and 1AKE, respectively) along with two intermediate conformations generated by our method. As expected, the LID and NMPbind domains change significantly compared to the CORE domain. Figure 6 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].

VI. Conclusions and Future Work

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 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 [42]. Finally, using T-RRT [43] 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 [44].

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.

REFERENCES

[1] A. F. Voter, "A method for accelerating the molecular dynamics simulation of infrequent events," *J. Chem. Phys.*, vol. 106, no. 11, pp. 4665–4677, 1997.

[2] S. Izrailev, S. Stepaniants, B. Isralewitz, D. Kosztin, H. Lu, F. Molnar, W. Wriggers, and K. Schulten, "Steered molecular dynamics," in *Computational Molecular Dynamics: Challenges, Methods, Ideas*. Springer-Verlag, 1998, pp. 39–65.

[3] R. R. Sørensen and A. F. Voter, "Temperature-accelerated dynamics for simulation of infrequent events," *J. Comput. Phys.*, vol. 112, pp. 9599–9606, 2000.

[4] A. Laio and M. Parrinello, "Escaping free-energy minima," *Proc Natl. Acad. Sci. U.S.A.*, vol. 99, no. 20, pp. 12 562–12 566, 2002.

[5] D. Hamelberg, J. Morgan, and J. A. McCammon, "Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules," *J. Chem. Phys.*, vol. 120, pp. 11 919–11 929, 2004.

[6] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, "Atomic-level characterization of the structural dynamics of proteins," *Science*, vol. 330, no. 6002, pp. 341–346, 2010.

[7] G. Mills and H. Jónsson, "Quantum and thermal effects in h2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems," *Phys. Rev. Lett.*, vol. 72, pp. 1124–1127, 1994.

[8] W. E, W. Ren, and E. Vanden-Eijnden, "String method for the study of rare events," *Phys. Rev. B.*, vol. 66, p. 052301, 2002.

[9] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, "Transition path sampling and the calculation of rate constants," *Annu. Rev. Phys. Chem.*, vol. 53, pp. 291–318, 2002.

[10] J. Cortés, T. Siméon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Siméon, and V. Tran, "A path planning approach for computing large-amplitude motions of flexible molecules," *Bioinformatics*, vol. 21, no. suppl 1, pp. i116–i125, 2005.

[11] B. Raveh, A. Enosh, O. Schueler-Furman, and D. Halperin, "Rapid sampling of molecular motions with prior information constraints," *PLoS Comput. Biol.*, vol. 5, no. 2, p. e1000295, 2009.

[12] N. Haspel, M. Moll, M. Baker, W. Chiu, and L. E. Kavraki, "Tracing conformational changes in proteins," *BMC Struct. Biol.*, vol. 10, no. Suppl 1, p. S1, 2010.

[13] I. Al-Bluwi, T. Siméon, and J. Cortés, "Motion planning algorithms for molecular simulations: A survey," *Comput. Sci. Rev.*, 2012, in press.

[14] Q. Cui and I. Bahar, *Normal mode analysis: theory and applications to biological and chemical systems*, ser. Chapman and Hall/CRC mathematical and computational biology series. Chapman & Hall/CRC, 2006.

[15] B. Brooks and M. Karplus, "Normal modes for specific motions of macromolecules: application to the hinge-bending mode of lysozyme," *Proc Natl. Acad. Sci. U.S.A.*, vol. 82, no. 15, pp. 4995–4999, 1985.

[16] K. Hinsen, "Analysis of domain motions by approximate normal mode calculations," *Proteins*, vol. 33, no. 3, pp. 417–429, 1998.

[17] F. Tama and Y. H. Sanejouand, "Conformational change of proteins arising from normal mode calculations," *Prot. Eng.*, vol. 14, no. 1, pp. 1–6, 2001.

[18] V. Alexandrov, U. Lehnert, N. Echols, D. Milburn, D. Engelman, and M. Gerstein, "Normal modes for predicting protein motions: a comprehensive database assessment and associated web tool," *Prot. Sci.*, vol. 14, no. 3, pp. 633–643, 2005.

[19] S. Kirillova, J. Cortés, A. Stefaniu, and T. Siméon, "An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins," *Proteins*, vol. 70, no. 1, pp. 131–143, 2008.

[20] B. P. Cossins, A. Hosseini, and V. Guallar, "Exploration of protein conformational change with PELE and meta-dynamics," *J. Chem. Theory Comput.*, vol. 8, pp. 959–965, 2012.

[21] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees : Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. Donald, K. Lynch, and D. Rus, Eds. Boston: A.K. Peters, 2001, pp. 293–308.

[22] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2008.

[23] F. Tama, F. X. Gadea, O. Marques, and Y. H. Sanejouand, "Building-block approach for determining low-frequency normal modes of macromolecules," *Proteins*, vol. 41, pp. 1–7, 2000.

[24] A. R. Atilgan, S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar, "Anisotropy of fluctuation dynamics of proteins with an elastic network model," *Biophys. J.*, vol. 80, no. 1, pp. 505–515, 2001.

[25] M. M. Tirion, "Large amplitude elastic motions in proteins from a single-parameter, atomic analysis," *Phys. Rev. Lett.*, vol. 77, no. 9, pp. 1905–1908, 1996.

[26] A. D. Schuyler and G. S. Chirikjian, "Efficient determination of low-frequency normal modes of large protein structures by cluster-NMA," *J. Mol. Graph. Model.*, vol. 24, pp. 46–58, 2005.

[27] O. N. A. Demerdash and J. C. Mitchell, "Density-cluster NMA: A new protein decomposition technique for coarse-grained normal mode analysis," *Proteins*, vol. 80, no. 7, pp. 1766–1779, 2012.

[28] E. Eyal, L. W. Yang, and I. Bahar, "Anisotropic network model: systematic evaluation and a new web interface," *Bioinformatics*, vol. 22, no. 21, pp. 2619–2627, 2006.

[29] J. Cortés and I. Al-Bluwi, "A robotics approach to enhance conformational sampling of proteins," *Proc. IDETC/CIE*, 2012, in press.

[30] M. Renaud, "A simplified inverse kinematic model calculation method for all 6R type manipulators," in *Current Advances in Mechanical Design and Production VII*, M. F. Hassan and S. M. Megahed, Eds. New York: Pergamon, 2000, pp. 57–66.

[31] H. Y. Lee and C. G. Liang, "A new vector theory for the analysis of spatial mechanisms," *Mechanism and Machine Theory*, vol. 23, no. 3, pp. 209–217, 1988.

[32] D. Manocha and J. F. Canny, "Efficient inverse kinematics for general 6r manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.

[33] E. A. Coutsias, C. Seok, M. P. Jacobson, and K. A. Dill, "A kinematic view of loop closure," *J. Comput. Chem.*, vol. 25, no. 4, pp. 510–528, 2004.

[34] A. Atramentov and S. M. LaValle, "Efficient nearest neighbor searching for motion planning," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 632–637, 2002.

[35] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain *et al.*, "The protein data bank," *Acta Cryst. D*, vol. 58, no. 6, pp. 899–907, 2002.

[36] D. A. Case, T. A. Darden, T. E. Cheatham III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, K. M. Merz, D. A. Pearlman, M. Crowley *et al.*, *AMBER 9*. San Francisco: University of California, 2006.

[37] E. Plaku, H. Stamati, C. Clementi, and L. Kavraki, "Fast and reliable analysis of molecular motion using proximity relations and dimensionality reduction," *Proteins*, vol. 67, no. 4, pp. 897–907, 2007.

[38] A. Shehu and B. Olson, "Guiding the search for native-like protein conformations with an ab-initio tree-based exploration," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1106–1127, 2010.

[39] C. W. Müller and G. E. Schulz, "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.*, vol. 224, no. 1, pp. 159–177, 1992.

[40] C. W. Müller, G. J. Schlauderer, J. Reinstein, and G. E. Schulz, "Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding," *Structure*, vol. 4, no. 2, pp. 147–156, 1996.

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

[42] D. Devaurs, T. Siméon, and J. Cortés, "Parallelizing RRT on distributed-memory architectures," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2261–2266, 2011.

[43] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–646, 2010.

[44] L. Jaillet, F. J. Corcho, J. J. Pérez, and J. Cortés, "Randomized tree construction algorithm to explore energy landscapes," *J. Comput. Chem.*, vol. 32, no. 16, pp. 3464–3474, 2011.