Enhancing backbone sampling in Monte Carlo simulations using Internal Coordinates Normal Mode Analysis

Victor A. Gil,[†] Daniel Lecina-Casas,[†] Christoph Grebner,[‡] and Victor Guallar[*,†,¶]

[†]*Joint BSC-CRG-IRB Research Program in Computational Biology, Barcelona Supercomputing Center, 08034 Barcelona, Spain*

[‡]*Department of Medicinal Chemistry, CVMD iMed, AstraZeneca, S-43183 Mölndal, Sweden*

[¶]*Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluís Companys 23, E-08010 Barcelona, Spain*

E-mail: victor.guallar@bsc.es

Abstract

Normal mode methods are becoming a popular alternative to sample the conformational landscape of proteins. In this study, we describe the implementation of an internal coordinate normal mode analysis method and its application in exploring protein flexibility by using the Monte Carlo method PELE. This new method alternates two different stages, a perturbation of the backbone through the application of torsional normal modes, and a resampling of the side chains. We have evaluated the new approach using two test systems, ubiquitin and c-Src kinase, and the differences to the original ANM method are assessed by comparing both results to reference molecular dynamics simulations. The results suggest that the sampled phase space in the internal coordinate approach is closer to the molecular dynamics phase space than the one coming from a Cartesian coordinate anisotropic network model. In addition, the new method shows a great speedup (∼5-7x), making it a good candidate for future normal mode implementations in Monte Carlo methods.

1 Introduction

Computational experiments are, in general, easier to prepare, and faster and cheaper to perform than their real-life counterparts. Furthermore, they allow scientist to have a privileged view of the systems under study, providing spatial and (often) timescale resolution that cannot be achieved by any other mean.

Of all the features that can be simulated in biopolymers, flexibility is one of the most difficult to reproduce and still remains a major challenge. The biological role of protein flexibility is highly relevant, being tightly related to protein function and ligand binding mechanisms, the understanding of which is a central aspect in computer-aided drug design. The correct handling of ligand and receptor flexibility is increasingly seen as critical for the success of virtual screening experiments.[1] To this aim, several approaches have been proposed to introduce flexibility: docking on multiple receptor conformations,[2,3] rotamer sampling on side chains and ligands,[4] molecular dynamics,[5] stochastic techniques,[6,7] etc.

The most known and used technique to sample protein flexibility, with great success, is molecular dynamics (MD). Long scale MD simulations can be used to describe protein folding or even to capture ligand binding events efficiently.[8,9] However, these simulations require large computational resources or specialized hardware, which limits their use. In order to be applicable in computer-aided drug design, the used sampling techniques need to be fast and efficient allowing for reasonable short timelines. This is also the main reason why a detailed handling of flexibility is greatly simplified in methods where speed is desirable, such as the ones typically used in virtual high-throughput screening (VHTS). Recent hardware improvements, especially in accelerators such as GPUs, are spreading the use of MD simulations in drug design projects,[10] however, they cannot be applied routinely in VHTS pipelines, where thousands of compounds have to be tested.

An alternative to MD are Monte Carlo (MC) techniques.[11] These methods have the capability of generating uncorrelated samples in two subsequent steps, allowing to quickly traverse the conformational space. This property permits, theoretically, to sample the conformational space more effectively and in a computationally tractable way. However, depending on the problem, the difficulty of generating new likely poses may largely drop its performance.[12] In particular, when modeling large biomolecules, MC methods have difficulties in efficiently sampling all degrees of freedom;[13,14] most successful MC studies (and methods) focus on small-medium polypeptides and/or local sampling, such as loops and side chains.[15,16] The Protein Energy Landscape Exploration (PELE) software[7] aims to overcome these sampling problems by adding protein structure prediction techniques in the MC sampling iteration.

PELE has shown to be successful in many tasks like sampling protein flexibility,[17] protein-ligand interactions,[18–20] enzyme engineering[21,22] and to describe ligand thermodynamics.[23]

1.1 The Anisotropic Network Model (ANM) in PELE

PELE is implemented as an iterative procedure where each MC iteration is composed of a perturbation and a relaxation phase. In the perturbation phase, the protein backbone is modified using an approach based on the ANM, a type of Normal Mode Analysis (NMA) first introduced by Atilgan and coworkers.[24] This model simplifies the protein potential as a harmonic elastic network of coarse grain units (typically C_α atoms) oscillating around a stable equilibrium conformation. These kinds of methods have shown to be successful in the study of large amplitude transitions in proteins, which are often involved in biological functions. Despite their simplicity, they have been shown to have good predictive capabilities and correlate well with experimental data.[25,26]

Once the elastic network has been defined and the Hessian ($H_{ij} = \frac{\partial^2 V}{\partial q_i \partial q_j}$) is calculated, eigenvectors and eigenvalues can be obtained by solving $HA = A\Lambda$.

Eigenvalues (Λ) are associated with the frequencies of the modes and the energetic cost of moving the system along its related eigenvector (A). Usually, only the first lower frequency modes are retrieved, as they are enough to describe wide conformational movements.[27] Keep in mind that higher frequency modes, associated with C_α -C_α stretching, do not have physical meaning. These lower frequency modes have shown to have a high degree of correlation with MD essential space.[28]

In order to apply the ANM modes in PELE, a subset of the lower energy eigenvectors is chosen (either randomly or as defined by the user) and a linear combination of them is calculated. The translation of the current coordinates by the resultant vector defines a set of target coordinates: the ANM conformation proposal. The conformational change is eventually calculated by adding harmonic constraints between the initial position of each C_α atom and their position in the proposal to finally perform a minimization that will move the atoms close to their target positions.

Because of the chosen strategy to apply the modes during the ANM step, the covalent geometry of the protein can be slightly distorted, producing large energy increments. In order to overcome this, PELE enters a relaxation stage where the energy is lowered. First, the torsional angles of the most energetic side chains are changed using a side chain prediction algorithm and a library of predefined rotamers. Second, a global minimization is performed in order to further lower the energy. In this system-wide minimization, a weak constraint is added to the C_α atoms to prevent the minimization from undoing the perturbation backbone move.

Finally, a metropolis acceptance step is performed using the Boltzmann criterion, and the current conformation is accepted or rejected. Thus, the overall approach is quite different from most MC implementations: each MC step involves a significant number of protein structure prediction techniques, requiring a remarkable amount of computational time ($\sim$1 minute on average), but is capable of introducing a large collective displacement. Notice that, by using these methods, we introduce importance sampling toward feasible structures (also adding limitations, see below).

One of the consequences of the use of minimizations is the loss of detailed balance, which implies that a complete exploration of all accessible states is not guaranteed. Also, we observe that, under certain circumstances, the combination of the profuse use of minimizations (twice every MC iteration) and the implicit solvent model adds a bias towards compact conformations.

1.2 Internal coordinates conformational sampling

Internal coordinates (IC) are a set of interdependent coordinates that include the distance, angle and dihedral angle between atoms, and are supposed to be a more natural way of representing and manipulating chemical entities.

Conformational sampling in torsional space is, in general, more efficient than in Cartesian coordinate (CC) space. The main reason is that changes in the torsional degrees of freedom often show low energy barriers, while bending and angular changes have higher energy barriers. This idea has often been used to enhance the sampling of MC algorithms.[29–32]

It is also possible to calculate normal modes in the internal coordinates space. Our preliminary analysis of torsional modes showed that they are more collective (see Section S1). This is an indicator of a less severe "tip effect", produced by a concentration of the modes in loose loops that can potentially nullify the movements in more structured regions, slowing down sampling. Besides, their degree of collectivity is less dependent on the topology of the elastic network (EN). When comparing them with CC modes, we have observed that both mode spaces do not always contain the same information (their overlaps range from 65% to 95%) possibly due to their differences in collectivity.

Internal coordinates NMA (icNMA) has already been used as a feasible alternative to handle protein flexibility. However, its presence in literature is incidental compared to its CC counterpart, maybe because the implementation of the first is harder. We would like to highlight the methods published by Noguti and Go,[33] Kidera *et al.*[32] and the more recent iMC method proposed by López-Blanco *et al.*[34]

In this article, we will focus on describing the improvements carried out on the PELE conformational sampling features; our long-term goal is to provide a fast and reliable backbone sampling technique to add into virtual screening refinement steps. To this end, we implemented an icNMA algorithm in PELE and adapted it to perform the backbone perturbation and thus handling protein flexibility. The resiliency to changes in the description of the elastic network, and the higher collectivity of IC modes, make them a perfect alternative for the current Cartesian coordinate NMA (ccNMA) step in PELE. Also, the way torsional changes are applied supposes a fundamental advantage compared to ccNMA: as it does not distort the covalent structure, the complex relaxation protocol is no longer needed. In order to understand the protein motion provided by the new method, as well as to assess its performance, we ran CC and IC NMA simulations for two different systems widely used in benchmarking: ubiquitin and c-Src kinase, and we compared our results with reference MD trajectories.

2 Materials and Methods

2.1 Internal coordinates normal mode analysis

The implemented IC NMA method starts by describing an elastic network of rigid units that encompass all the heavy atoms among rotatable backbone torsions. This means that two rigid units per residue are defined instead of only one as described by ANM (see Fig. 1A).

The potential of this spring network is the sum of all Hookean interactions between the units. If expressed using generalized internal coordinates (backbone dihedrals in this case) the potential can be written as:

$$V = \frac{1}{2}(q - q^0)H(q - q^0)^T, \tag{1}$$

where H is the Hessian, that in terms of q[35] can be written as:

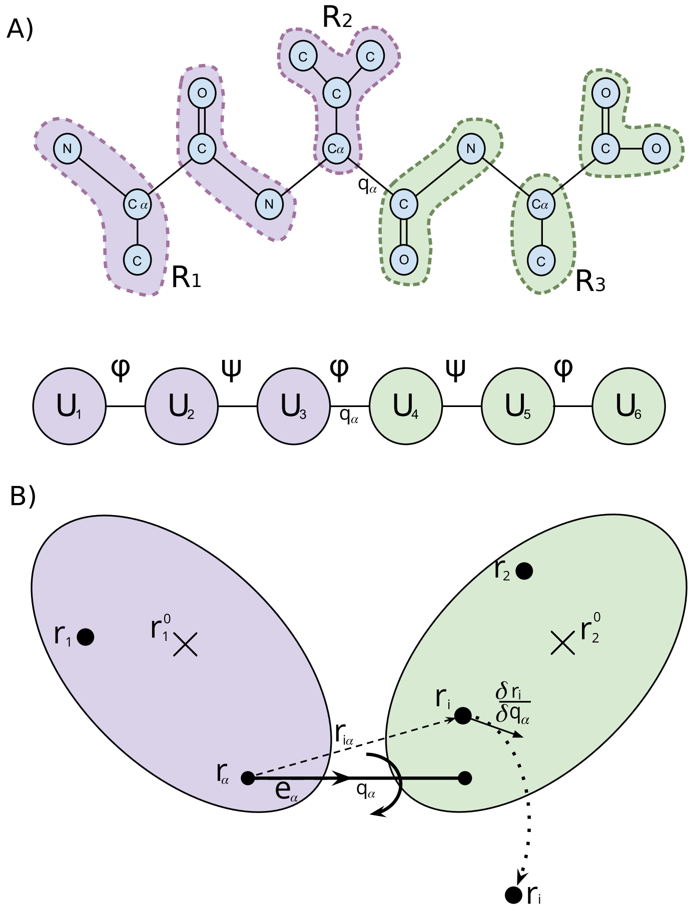

Figure 1: A) The coarse grain model defines units encompassed by the torsion angles *phi* and *psi*. B) Schematic representation of the rotation of units 4-6 (green) around torsion q_α, including the notation used in the formulae.

$$H_{\alpha,\beta} = \frac{\partial^2 V}{\partial q_\alpha \partial q_\beta} = \sum_{i<j} \frac{f_{ij}}{|r_{ij}|^2} \left\langle r_i j, \frac{\partial r_i - \partial r_j}{\partial q_\alpha} \right\rangle . \left\langle r_i j, \frac{\partial r_i - \partial r_j}{\partial q_\alpha} \right\rangle . \tag{2}$$

By imposing Eckart conditions[36] and that the origin of the molecule is the center of mass, Noguti and Go proposed an analytical solution for the partial derivatives[37] so that

$$\frac{\partial r_1}{\partial q_\alpha} = e_\alpha \times \left(\frac{M_2}{M} r_\alpha + \frac{M_1}{M} r_1^0 \right) - r_1 \times \frac{M_1 r_1^0 \times (e_\alpha \times r_\alpha) + I_2 e_\alpha)}{I} \tag{3}$$

$$\frac{\partial r_2}{\partial q_\alpha} = -e_\alpha \times \left(\frac{M_1}{M} r_\alpha + \frac{M_2}{M} r_2^0 \right) + r_2 \times \frac{M_2 r_2^0 \times (e_\alpha \times r_\alpha) + I_1 e_\alpha)}{I}, \tag{4}$$

where symbols without subscript refer to global quantities and symbols with superscript refer to the set of units to the left (1) or to the right (2) of the rotation axis, M is the mass, I the inertia, r^0 the center of mass and r are the atom positions. These derivatives describe how Cartesian coordinates change upon rotation around the e_α axis (torsion q_α), in such a way that the momentum is conserved (see Fig. 1B).

The recursive method proposed by Noguti and Go[33] and by Abe and coworkers[38] lowers memory consumption as well as computational complexity, which decreases from $\theta(n^4)$ to $\theta(n^2)$. The kinetic energy can than be expressed in terms of the generalized coordinates q:

$$K = \frac{1}{2} \dot{q}^T K \dot{q} \tag{5}$$

and the metric tensor K:

$$K_{\alpha,\beta} = \frac{\partial^2 K}{\partial \dot{q}_\alpha \partial \dot{q}_\beta} = \sum_i^n m_i \left\langle \frac{\partial r_i}{\partial q_\alpha}, \frac{\partial r_i}{\partial q_\beta} \right\rangle , \tag{6}$$

which can be calculated as:[37]

$$\begin{gathered} K_{\alpha\beta} = \frac{M_1 M_3}{M} \left[e_\alpha \times (r_\alpha - r_1^0) \right] \left[e_\beta \times (r_\beta - r_3^0) \right] + \\ \left[M_1 r_1^0 \times (e_\alpha \times r_\alpha) - I_1 e_\alpha \right] I^{-1} \left[M_3 r_3^0 \times (e_\beta \times r_\beta) - I_3 e_\beta \right] \end{gathered} \tag{7}$$

The eigenvectors and eigenvalues are obtained solving the eigenproblem $HA = KA\Lambda$. On this occasion the meaning of a eigenvector is no longer a displacement in Cartesian coordinates, but a set of differential rotations around the ϕ and ψ torsions of the protein backbone. It is worth noting that, as torsional NMA uses less degrees of freedom than ANM (∼2 vs. 3 per residue), the Hessian matrix is smaller and therefore its diagonalization is faster. However, this is not going to affect the overall performance of the proposed method, as frequently modes are calculated only few times along the simulation.

2.2 IC-based sampling method

The new implemented method can be divided in two independent stages: backbone perturbation and side chain perturbation, both implemented as MC algorithms. Each iteration of the backbone perturbation stage (icNMA step) is analogue to an ANM step in regular PELE (ccNMA step). First, the increments for ϕ and ψ torsions of the backbone are calculated using the eigenvectors. Then, the resulting differential rotations are rescaled using the maximum amplitude chosen from a user-defined range $[a_{min}, a_{max}]$. The value is drawn from a normal distribution with mean $^{a_{min}+a_{max}}/_{2}$ and standard deviation $^{a_{min}-a_{max}}/_{4}$. The distribution is truncated so that, when the chosen amplitude is outside the range, the draw is repeated. The angular increments are applied using Choi's method.[39]

In general, small torsional displacements can give place to large linear displacements, favouring the appearance of steric clashes. The energy increment caused by the clashes will most likely ensure rejection of the backbone move, decreasing acceptance dramatically. As this issue is especially evident in side chains, we have chosen to minimize the energy of those side chains whose atoms collide with other atoms of the system, using PELE implementation of a Truncated Newton minimizer.[40] While this minimization breaks detailed balance, the following stage (see below) aims at recovering proper side chain sampling.

The side chain perturbation stage, also implemented as an MC algorithm, is performed right after the backbone perturbation stage. At each iteration, a residue with a side chain

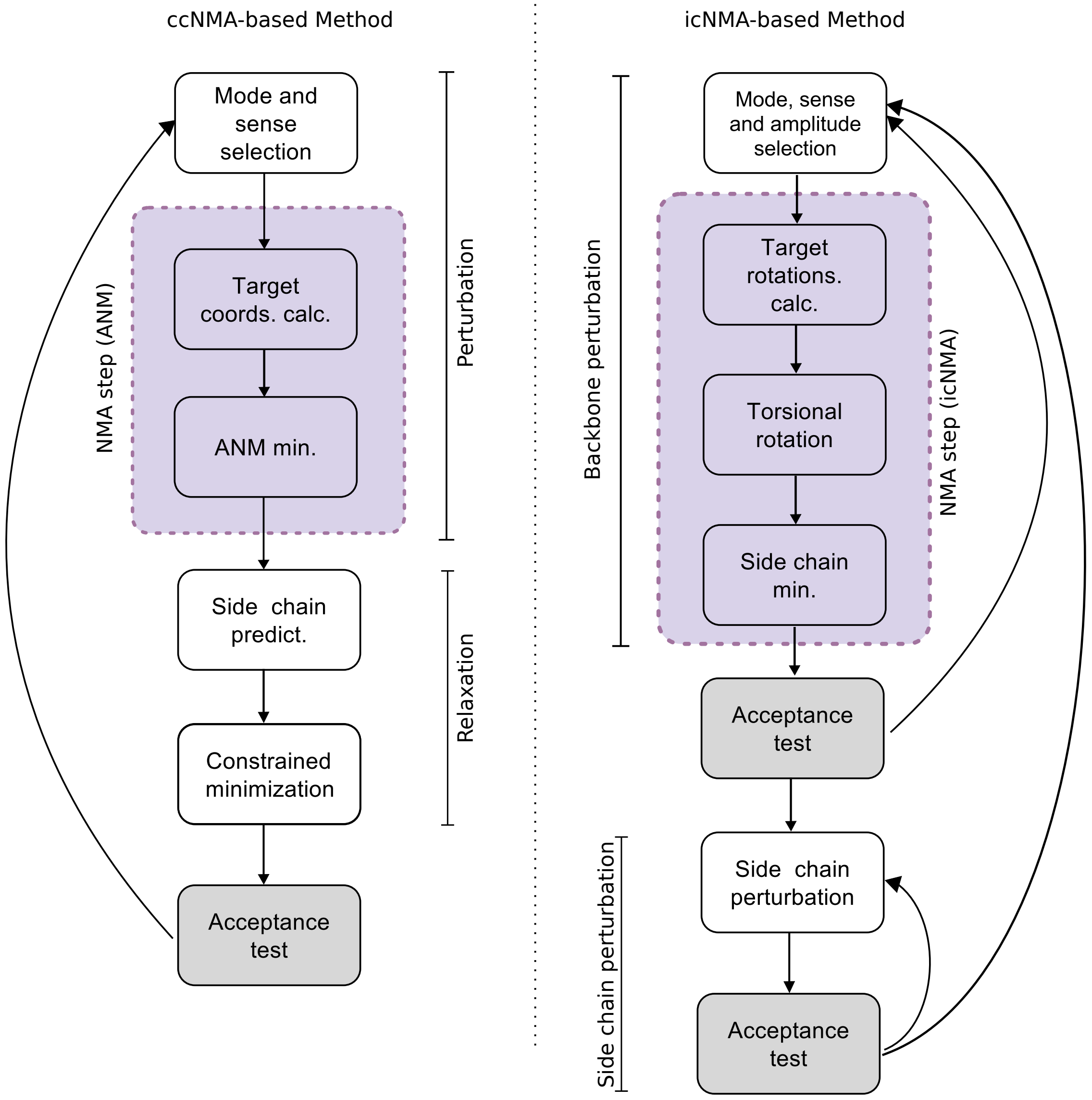

Figure 2: Diagram of the two methods currently implemented in PELE. In the ccNMA-based method, each iteration is composed of two parts: the ANM perturbation and the relaxation. The new icNMA-based method consists of two independent perturbation stages, both implemented as MC algorithms. In the first stage (backbone perturbation), the backbone is modified, in the second (side chain perturbation), a proper sampling of the side chains is recovered.

with rotatable bonds is randomly chosen. Then a random increment for each of its rotatable bonds is sampled from a truncated normal distribution (defined as before), a sense for the rotation is randomly selected, and the new side chain conformation is built. After this, the energy increment of the system is calculated and the proposal is accepted or rejected depending on the outcome of the Metropolis test.

2.2.1 Test systems and reference simulations

We have decided to use MD as a reference in order to compare the two backbone sampling methods. MD simulations are a standard approach for investigating protein dynamics, even though assessing full convergence is hard to achieve.[41] We assume that a long enough MD simulation (i.e. beyond the μs) should be capable of sampling the relevant conformational space.[42,43]

We have selected Ubiquitin (PDB id: 1UBQ) and c-Src-kinase (PDB id: 1Y57, residues 258:534) as test systems because they show very different structural and dynamic properties. Ubiquitin is a small globular protein with only 76 amino acids which can be found in nearly all eukaryotic cells. It plays a crucial role in post-translational modifications and modifies the function of substrate proteins.[44–46] It is very well investigated, both experimentally[47–52] and theoretically.[53–55] The structure is well characterized by X-ray (PDB-id: 1UBQ[47]) and NMR (e.g. PDB-id: 2MSG[52]). This makes it an ideal test system for new methods and it is often used as part of benchmark sets.[56–58] The reference MD simulation for ubiquitin is taken from a previous benchmark of PELE.[59] The trajectory was calculated with GROMACS 4.0.5[60] using the OPLSAA force field[61] and explicit solvent (a cubic box of water molecules). It was run at 300 K and 1 atm, using periodic boundary conditions. The production run was performed for 1 μs. We analysed one snapshot every 100 ps. In the PELE simulations, we omitted the last three residues of the n-terminal loop, as its high flexibility is a source of “tip effect” and masks the flexibility of the rest of the protein.

c-Src is a cytoplasmic tyrosine kinase that catalyzes the transfer of a phosphate group from ATP to the hydroxyl group of the tyrosine residue. Src plays an important role in cellular proliferation, survival, migration, and angiogenesis.[62] It shows wide inter-domain displacements between its three well-defined domains, the N-lobe, C-lobe and hinge. Our reference MD simulation for c-Src is a 20 μs MD simulation describing the binding of PP1 (pyrimidine-type) inhibitor, a Src-selective tyrosine kinase inhibitor binding to the ATP-binding site of the kinase,[63] presented by Shan *et al.*[8] It was parameterized using a corrected[64,65] Amber99SB force field and the simulation was run using Desmond 2.2.2.1 software[66] in the Anton specialized hardware[67] with explicit water solvent. Although the simulation was performed in the presence of a ligand, we assume that only the populations of visited states[68,69] change, but the accessible conformational space does not change considerably.

2.3 Comparison analyses

In order to assess the performance of the different approaches, we carried out different analyses and compared them against our reference classical MD simulations. The chosen analyses are:

2.3.1 Root mean square fluctuation (RMSF)

The RMSF is an isotropic measure of the displacement over time of an atom from a reference position (typically the average structure after superposition of all frames). It is a common measure of conformational sampling efficiency, however it cannot be used as the only source to compare different methods, as the exploration of very different zones could lead to the same RMSF. It can be calculated as

$$RMSF \equiv \sqrt{\langle (x - \langle x \rangle)^2 \rangle} = \sqrt{\frac{1}{F}\sum_{n=1}^{F}(x(n) - \langle x \rangle)^2} \tag{8}$$

where F is the number of structures of the conformational ensemble, and $\langle x \rangle$ is the average position.

2.3.2 Solvent-accessible surface area (SASA)

The SASA describes the accessible surface for a spherical probe which is rolling over the molecule of interest. The probe, with a typical radius set to 1.4 $\mathring{A}$, mimics a water molecule, giving a measure of the exposition of the molecule to the solvent and, therefore, its compactness.[70]

2.3.3 Radius of gyration

The radius of gyration is a measure of the dispersion of a set of atoms to its center of mass.

$$R_g \equiv \sqrt{\langle (r - r_{COM})^2 \rangle} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - r_{COM})^2} \tag{9}$$

Thus, it is a measure of its compactness.[71] By measuring the radius of gyration at different times, one can assess protein conformational changes. Both SASA and the radius of gyration have been calculated using the VMD software.[72]

2.3.4 Conformational Space Overlap

In order to perform a purely geometrical comparison of the ensembles produced by each method, it is convenient to calculate the overlap between the conformational space of the reference MD and the normal mode driven trajectories. We used a similar approach to the ones shown in Lymann's and Lindorff-Larsen's articles:[73,74]

- First, we want to obtain a partition in k regions of our reference conformational space $(R_1, R_2, \ldots, R_k)$. To this end, we use the cluster analysis software pyProCT[75] and apply a k-medoids algorithm using the RMSD between conformations as the distance. The medoids of each of these clusters $(r_1, r_2, \ldots, r_k)$ are the representative conformations of each region R_i of the conformational space.

- Then, for each conformation c_j belonging to the CC or IC NMA ensembles, we look for its most similar MD representative r_i and we add it to the cluster R_i. In other words, this means that conformation c_i is assigned to the region R_i of the conformational space.

- For all the methods, we calculate the population of all the clusters. Afterwards, values are normalized in order to get a probability density distribution.
- Finally, the square root of the Jensen-Shannon divergence (JSD) of the distributions is calculated as

$$O = JSD(P||Q)^{\frac{1}{2}} = \left(\frac{1}{2}KLD(P||M) + \frac{1}{2}KLD(Q||M)\right)^{\frac{1}{2}}, \tag{10}$$

  where KLD is the Kullback-Liebler divergence and Q is the average distribution. The square root of JSD can be used as a metric[76] to assess the degree of overlap of the probability distributions, and, therefore, of each one of the methods with MD. Values are confined into the range [0,1], where a value of 1 means that the population distributions diverge completely and a value of 0 that the distributions are analogous.

2.3.5 Computational efficiency of deformation

Another goal of this work is to assess the temporal performance of the new method. We are mainly interested in two aspects, the time needed to complete an iteration, and the amount of deformation produced in each iteration (calculated as the RMSD of the structures before and after the iteration). We will combine both measures to calculate the computational efficiency of the deformation process as ${}^{RMSD_{step}}/{}_{t_{step}}$, the time needed to deform the backbone 1 $\mathring{A}$.

3 Results and discussion

3.1 Simulation setup

Both normal mode methods are customizable, letting the user control most aspects of the simulation. Since we are introducing a new methodology, we tuned the values of some simulation parameters in order to obtain the best performance and to compare both methods. First, we set the values of the common parameters like the temperature (300 K), or the cutoff distance of the elastic network (set to a 9 $\mathring{A}$, a value that is in agreement with the literature[77] and produces those highly collective modes; see Section S1). We decided to calculate up to 10 modes and to not combine them linearly. The mode and the sense of the movement are randomly changed at each MC iteration, which is a general and system-agnostic solution. Also, we chose to calculated the modes once at the beginning of the simulation, assuming that the potential surface does not change too abruptly.

Second, we determined the most sensitive parameters for each method. The first one, the displacement magnitude, controls the maximum translation ($\mathring{A}$) or the maximum torsional rotation (rad) performed in the NMA step. The second one, the minimum root mean square gradient (RMSG), is the threshold of convergence of the minimizations and can be understood as the "strength" of the minimization. In the ANM method, it modules the closeness of the final conformation to the proposal, whereas in the IC NMA method, it regulates the intensity of the side chain minimization, and it is related to the success releasing steric clashes.

Third, we performed a first round of simulations setting these parameters with all the combinations of the values found in Table 1. In order to find the best settings, we discarded the results with acceptances out of our working range (20-40%[78]) and we ranked them according to the similarity of their RMSF profiles with the reference MD simulation and favouring large backbone deformations. Then, we performed a second round of simulations, fine tuning the best displacements found in the previous round.

For the ubiquitin system, we found that the optimal values for our working parameters were a displacement magnitude of 1.08 $\mathring{A}$ and an RMSG of 0.1 $^{kcal}/_{mol\mathring{A}}$for the CC simulations, and a displacement magnitude in the range of 0.07 to 0.16 rad and an RMSG of 0.1 $^{kcal}/_{mol\mathring{A}}$for the IC simulations. Likewise, for c-Src kinase, the best values for the CC simulations were 0.66 $\mathring{A}$ for the displacement magnitude and and 0.1 $^{kcal}/_{mol\mathring{A}}$for the RMSG. A displacement range of 0.07 to 0.14 rad and RMSG of 0.05 $^{kcal}/_{mol\mathring{A}}$was found to be the best fitted parameters for the icNMA simulation.

As for the side chain perturbation stage in the icNMA method, we chose to perform 2000 side chain changes every ten steps of icNMA. In this case, we performed dihedral rotations in the range of 0.02 to 0.024 rad. These values are again optimized so that they yield acceptances between 20 and 40%.

Finally, we performed 12 independent 24 h production runs for each method and system.

Table 1: Choice of parameters affecting the mode application step in both studied methods and the values that will be used in the characterization tests.

Method	Displacement magnitude ($\mathring{A}$ rad)	Relaxation strength ($^{kcal}/_{mol\mathring{A}}$)
ccNMA	0.25, 0.66, 1.08, 1.5, 1.92	0.01, 0.02, 0.04, 0.05, 0.1
icNMA	0.02, 0.05, 0.075 ,0.10, 0.12, 0.15	0.01, 0.02, 0.04, 0.05, 0.1

3.2 Energetic cost of the NMA perturbation

We calculated the energy increments of the NMA step in the IC simulations as well as the energy increments of the ANM step and full iteration (perturbation+relaxation) of the CC simulations. The results show that icNMA is able to make MC proposals that are energetically favorable in both systems (see Table 2 and 3), while the ccNMA-based method cannot (the energy increments of this step are almost always positive). Without a way to make these energies decrease, it would be almost impossible to accept any step. It is only thanks to the relaxation phase of the CC algorithm, that the energy of the MC proposals can be lowered. It is worth noting that the IC method can result in large increments in energy (considerably larger than in CC) due to unresolved backbone clashes from dihedral rotation. It happens more often in ubiquitin than in the c-Src kinase, since the first is more globular than the latter, and steric clashes can be introduced more easily.

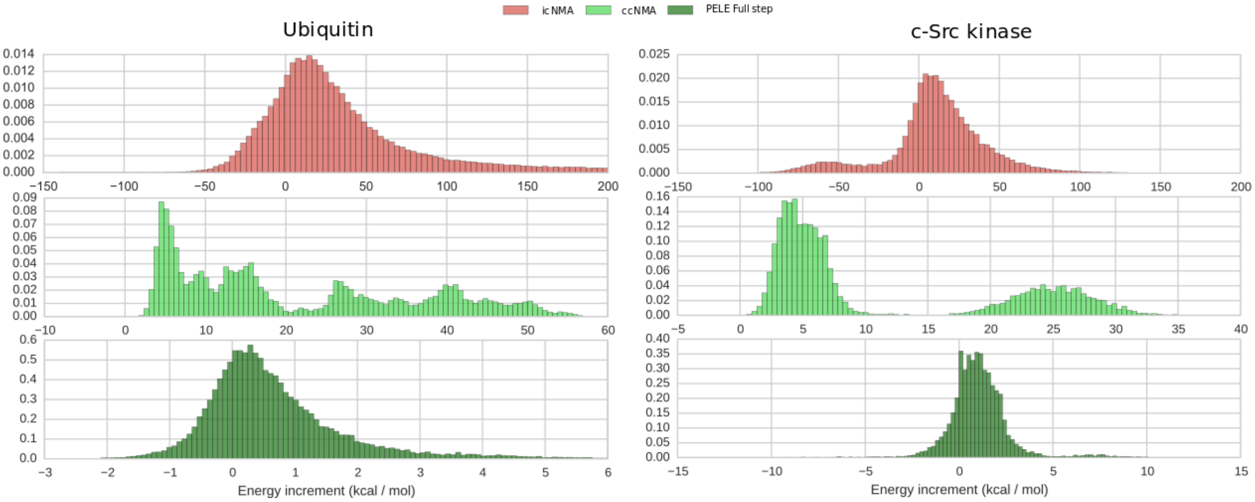

Figure 3: Distribution of the energy increments produced on each iteration for each method and protein system. In red, we show the energy increments in the icNMA step. In light and dark green we show the distributions of energy increments related to the ccNMA step and the complete PELE step, respectively. Note the changes in axis scales.

Table 2: Acceptance and percentage of energetically favorable proposals (EFP) generated by each algorithm in our simulations. In the CC, we can find two values: the first value belongs to the EFP of the ANM step, the second, between parentheses is the EFP of the whole PELE step (i.e. including the relaxation phase).

	Ubiquitin		c-Src kinase	
	CC	IC	CC	IC
Acceptance (%)	45.3	19.4	37.5	28.2
EFP (%)	<0.1 (26.6)	18.7	<0.1 (21)	27.2

3.3 Compactness of the protein (SASA and radius of gyration)

As it can be seen in Fig. 4, both ensembles generated by NMA-based methods are more compact than the ones obtained using MD. Comparing both normal mode methods, we see

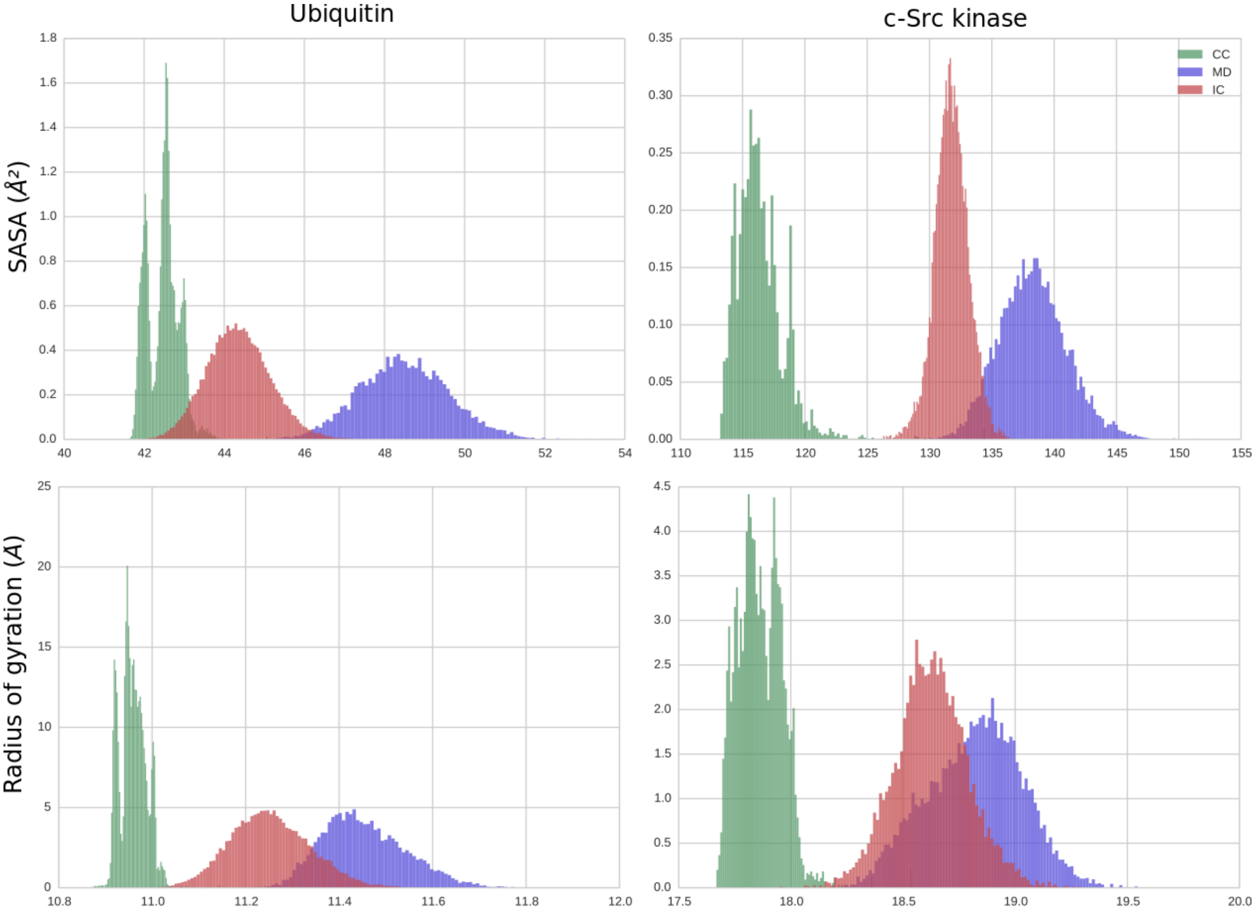

Figure 4: SASA and radius of gyration distributions for all the studied ensembles and systems. In general the measures obtained from the ensembles produced by the IC method look similar to our reference MD simulations.

that protein structures obtained by applying icNMA modes have larger SASA values and, therefore, seem to be less compact and collapsed than those obtained using the ccNMA method. We can draw similar conclusions by looking at the radius of gyration distribution.

We could expect differences, since the NMA-based methods are using an implicit solvation model (the Onufriev, Bashford, and Case, OBC[79] model), whereas the MD simulations were run using explicit solvent. The bias of some implicit solvent models, including OBC, to compact structures is well studied.[79–83] This bias is mainly caused by the over-stabilization of nonpolar interactions[84] and the increase in number and stability of hydrogen bonds.[85] This effect is further emphasized in the CC model by the two minimization procedures (see below). In current PELE simulations this is typically avoided by adding a weak (∼0.1 $^{kcal}/_{mol\mathring{A}^2}$) harmonic constraint every 10 alpha carbons, for example, to the initial model.

3.4 Protein fluctuations (RMSF)

RMSF plots of the ubiquitin simulations show small overall fluctuations for all methods (see Fig. 5A), between 1 and 2 $\mathring{A}$. This is expected, as ubiquitin is a relatively stiff protein. The movement is concentrated on the less structured parts: the loops between beta-sheets and alpha-helices (see Fig. 5D). In general, icNMA fluctuations are in closer agreement with MD than ccNMA fluctuations and also show a higher baseline. The scaled RMSF (see Fig. 5B) allows us to have a clearer view of the relative magnitudes of fluctuations. The β2-α-helix loop is more stable in the MD and IC simulations, while it shows a prominent peak in the RMSF for the CC simulation. This peak belongs to a temporary backbone rearrangement that is not present in the IC and MD simulations. CC clearly populates a further state with larger RMSD than IC and MD (See Fig. 5C). However, ccNMA clearly underestimates the movement of the β-4-β5 loop, which is nicely captured by icNMA.

Our c-Src kinase simulations are again producing smaller overall fluctuations than MD (with the exception of the P-loop). The rescaling of the RMSF plots shows that the ANM method is able to capture the flexibility of the A-loop with slightly more success (see 6 B).

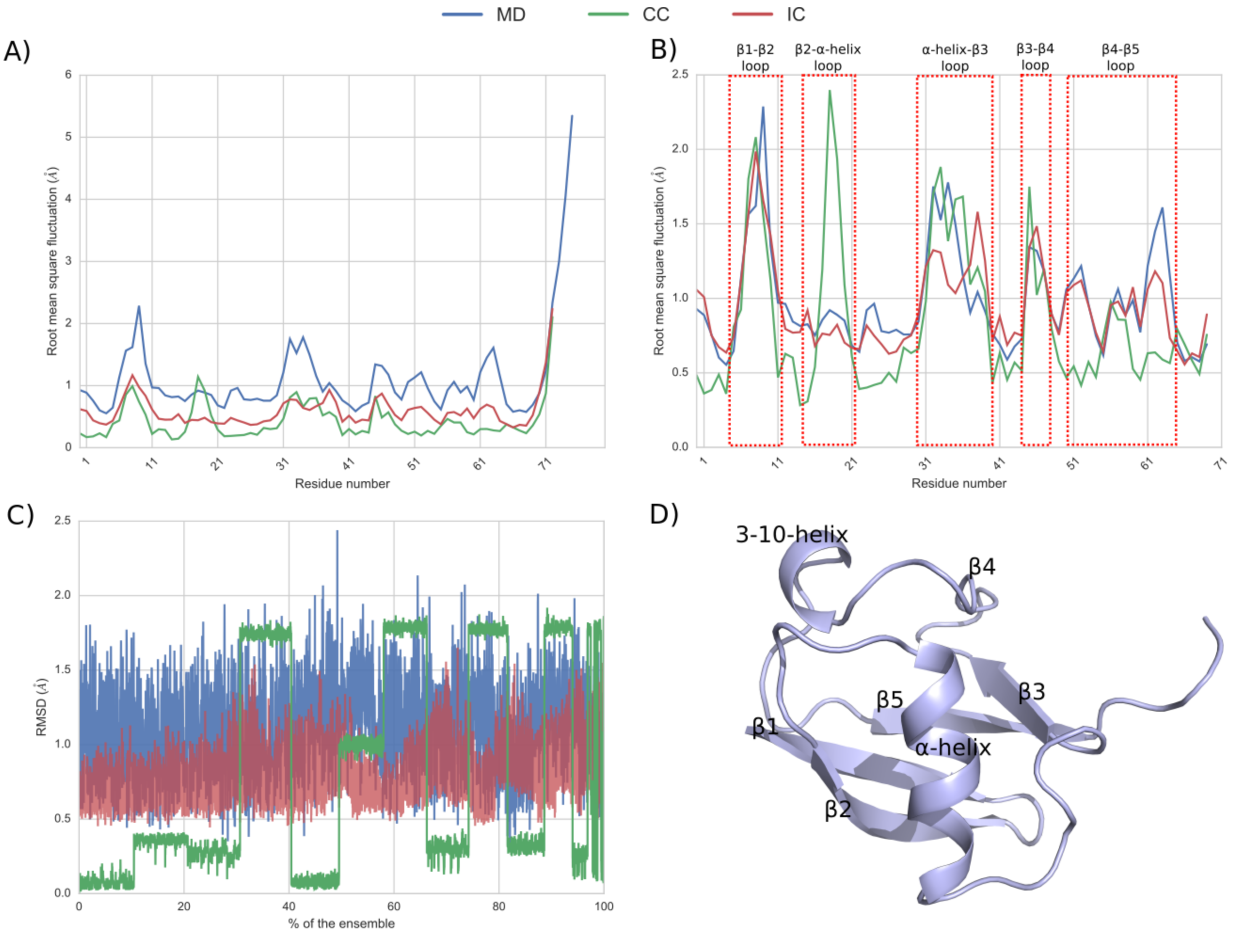

Figure 5: A) RMSF profile of the ubiquitin NMA simulations compared with MD. The last three residues have been excluded in order to milden the “tip effect”. B) We have superimposed the RMSF plots so that relative fluctuations can be checked. In order to do this, we have scaled them so that the root mean square error was minimum. This yielded scaling factors of 2.10 (CC) and 1.7 (IC). The most flexible parts of the protein (mainly the loops connecting the secondary structure) have been highlighted in this scaled representation of the RMSF. C) RMSD plot of the $\beta 2$-α-helix loop referred to the first CC simulation frame. D) Representation of ubiquitin identifying its secondary structure elements.

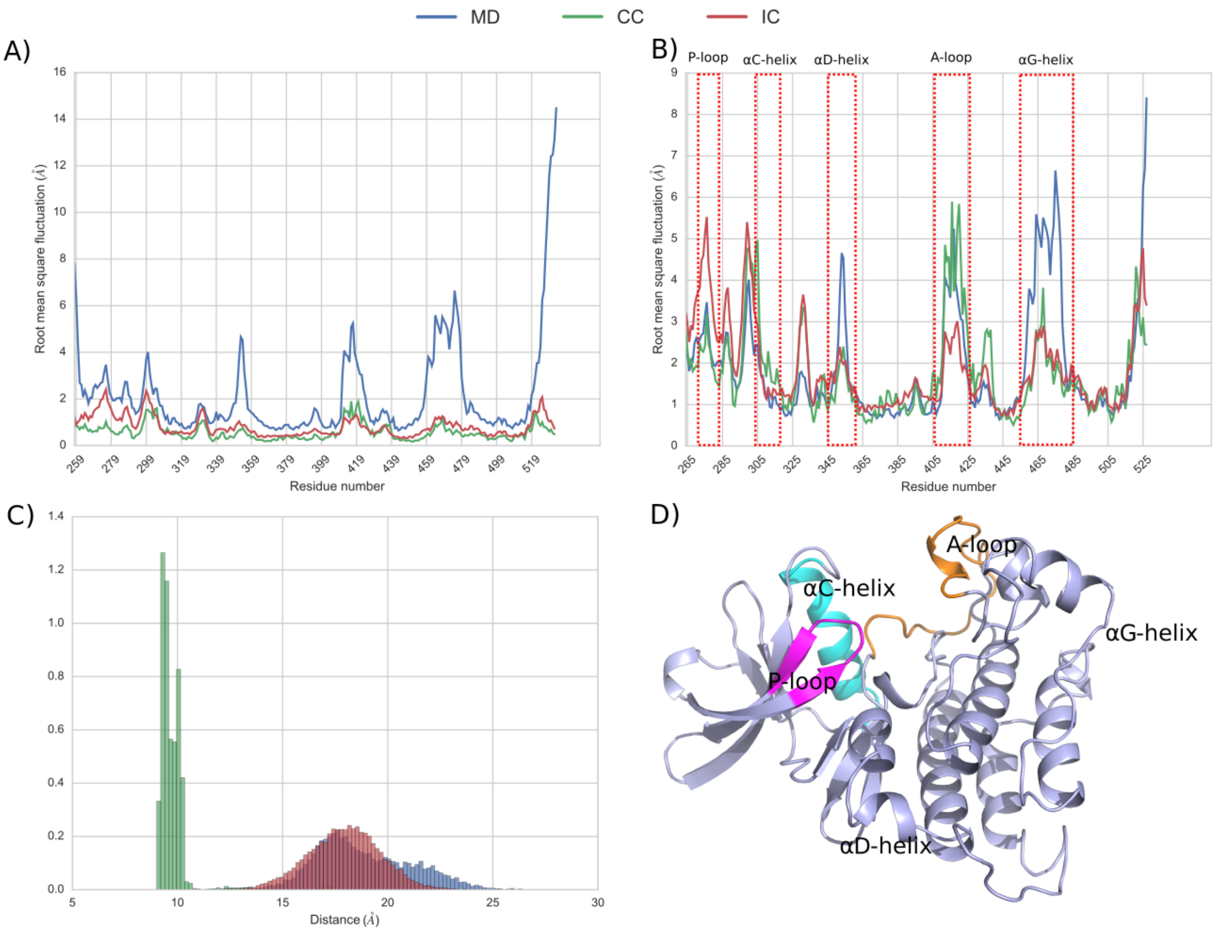

Figure 6: A) RMSF plot of the c-Src kinase NMA simulations compared with MD. N and C-terminal loops are omitted B) Scaled RMSF plot with highlighted functionally relevant elements. The scale factors are 3.10 (CC) and 2.3 (IC). C) Distributions of the distances between CYS:277:CA and LEU:387:CA for each simulation. D) Representation of the protein identifying its more relevant secondary structure elements.

Major differences can be found in the fluctuations of the residues belonging to the αD-helix, A-loop and αG-helix (see 6A, B and D). These structures show wide local rearrangements, in essence folding-unfolding events, that are hard to capture using NMA-based techniques.

The correct sampling of the P-loop dynamics is of utmost importance, as it is directly involved in the binding process.[86] We decided to investigate the opening and closing of this loop in more detail by measuring the distance between a central residue in the P-loop (CYS:277:CA) and a second residue on the other side of the binding site (LEU:387:CA) (see Fig. 6C). This distance shows a similar fluctuation range in the MD and icNMA simulations ($\sim$15-25 $\AA$ and $\sim$13-23 $\AA$ respectively), whereas the range of distances sampled by the ANM-based simulation is significantly smaller ($\sim$9.5-10.5 $\AA$). This is related to the increase of compactness observed in the SASA and rgyr analyses: the protein collapses quickly and the inter-domain distance does not oscillate much (see Fig. 4).

The lower overall fluctuations of both normal mode methods, as well as the low baselines, suggest that the proteins are moving less. This is partly due to the lack of local motion which can only be mapped with higher frequency modes, not present in our simplified NMA procedure. In addition, in the icNMA method there are no anharmonic backbone movements, further limiting its comparison with MD; anharmonicity has been studied to play an important role in the modulation of the amplitude of fluctuations.[77,87]

3.4.1 Effect of the minimization in the compaction process

In order to gain more insight on the compaction process and the effect of minimizations on it, we study how the distance between CYS:277:CA and LEU:387:CA changes in the different stages of the algorithms in the closing regime. It is convenient to define $r_d \equiv \frac{\sum_{i \in D_-} d_i}{\sum_{i \in D_+} d_i}$, where D_+ and D_- are the domains defined by positive and negative increments respectively, and d_i are the distance increments.

We will focus on the three parts of the ccNMA-based algorithm related to the change of the backbone (see Figure 2): the target coordinates calculation, where we make a proposal, the ANM minimization, where we apply the minimization with spring constraints toward the proposals; and the system-wide constrained minimization. In the icNMA-based algorithm, we will focus on the target rotation calculation (i.e. proposal rotations) and the torsional rotation, where we apply the rotation.

In the proposal stage (see Table 3), $r_d \simeq 1$, since proposals are symmetric by construction (random choice of modes and senses). In the application of the ANM modes, there is symmetry again, which means that the ccNMA step is successfully generating conformations close to the proposal. However, the constrained minimization shows a recurrent bias toward negative increments ($r_d \simeq 1.23$), leading to more compact structures. In the icNMA-based algorithm, proposals are again symmetric by construction, and torsional rotations are set to those values, giving no overall bias ($r_d \simeq 1$).

This agrees with the results previously seen in Figure 4, where ccNMA generated structures tend to be more compact, since backbone movement is minimization-driven. That would explain the tendency to over-close the protein, whereas in the icNMA step this does not happen, since minimizations do not apply to the backbone. These results demonstrate that the use of minimizations plays an important role in the compaction process.

Table 3: r_d values in the different stages of the ccNMA and icNMA-based methods.

	CC	IC
Target coordinates	0.98	0.95
ANM application	1.02	0.95
Constrained minimization	1.23	

3.5 Conformational space overlap

We have applied the algorithm described in Section 2.3.4 for partitions of 1 to 1k clusters. In panel A of Figure 7, we can see that, for all the studied numbers of clusters, the JSD is always lower for icNMA than for ccNMA. This means that the exploration of the configurational space performed by icNMA is in closer agreement with MD. The JSD can tell us whether NMA methods are populating the same regions as MD trajectories, however it does not give us information about the structural similarity of the conformations. In order to analyze it, we have calculated the average C_α RMSD of each structure in the NMA ensembles with the most similar MD conformation inside its cluster (see lower subplots of Fig. 7A). The RMSD results for ubiquitin are similar for both methods, showing an average RMSD value of ∼0.8 $\mathring{A}$. In the c-Src kinase case, however, results are significantly different between methods: the RMSD difference of 0.8 $\mathring{A}$ is indicating that icNMA is not only populating similar regions of the space, but also generating similar conformations to MD. Taking $k = 10$ as a case study (see Fig. 7B), we can observe that:

Ubiquitin Both NMA methods are visiting only a fraction of the possible regions of the conformational space sampled by MD (3 out of 10). The exploration performed by the ccNMA method is clearly even less than the one performed using the icNMA method: more than 90% of the population is concentrated in one cluster. This can be explained with a quick look to the RMSD matrix of the combined ensembles (Fig. 7C) : the submatrices of the NMA methods are more similar between themselves than to the MD submatrix. This is a consequence of the NMA methods failing to model the flexible loops connecting the secondary structure.

c-Src kinase The populations of the icNMA ensemble is distributed more evenly than the ccNMA ensemble populations. However, the icNMA method is still overpopulating one state (∼60% of its population). The ccNMA algorithm is visiting only 2 regions, a poor result compared with the 7 out of 10 regions that the icNMA is able to populate. In the subplot, we depicted the average interdomain distance for the structures of each cluster. icNMA has better agreement with MD, capturing the different stages of the P-loop opening/closing process. The RMSD matrix shows, again, notable differences between the ccNMA and the MD ensembles. However, the differences between icNMA and MD are less noticeable, and this can be related to the successful modelling of the

P-loop behaviour .

3.6 Computational performance of the methods

We have measured the time required to complete different tasks: an icNMA step, a ccNMA step, and a full PELE iteration (ccNMA step + relaxation phase). We have also calculated the extent of the perturbations performed in each task by calculating the RMSD of the structure before and after the task. We observe that the average time required to perform a ccNMA step is lower than the average time needed to perform an icNMA step (see Table 4), however, their efficiency distributions are pretty similar (see Fig. 8). This indicates that, although the icNMA step is slightly slower, the perturbations performed are wider than the ones applied by the ccNMA step. Nevertheless, the perturbations performed in the ccNMA step are, in general, not energetically favorable (already seen in Section 3.2), which forces PELE to add a relaxation phase. If we take into account the time needed to run the relaxation phase, the IC methodology clearly outperforms the CC methodology (the speedups per step are ∼5x and ∼7x, depending on the system). This explains the dramatic differences in their efficiencies illustrated in Fig. 8.

As the relaxation phase is an essential part to the ccNMA-based algorithm, so is the side chain perturbation stage to the icNMA method. In order to study the impact of this stage on the overall time, we have added the time needed to run 200 iterations of side chain perturbation to the icNMA time (remember that the ratio of iterations between the first and second stages is 10:2000). As expected, the resultant distributions show a shift to the left (less efficiency). This effect is more significant in the ubiquitin case than in the c-Src kinase case, mainly because the amortized side chain perturbation time (∼2 to 3s) is of the order of the time spent in an icNMA iteration. However, in both cases, it still shows a better overall efficiency than the CC-based method.

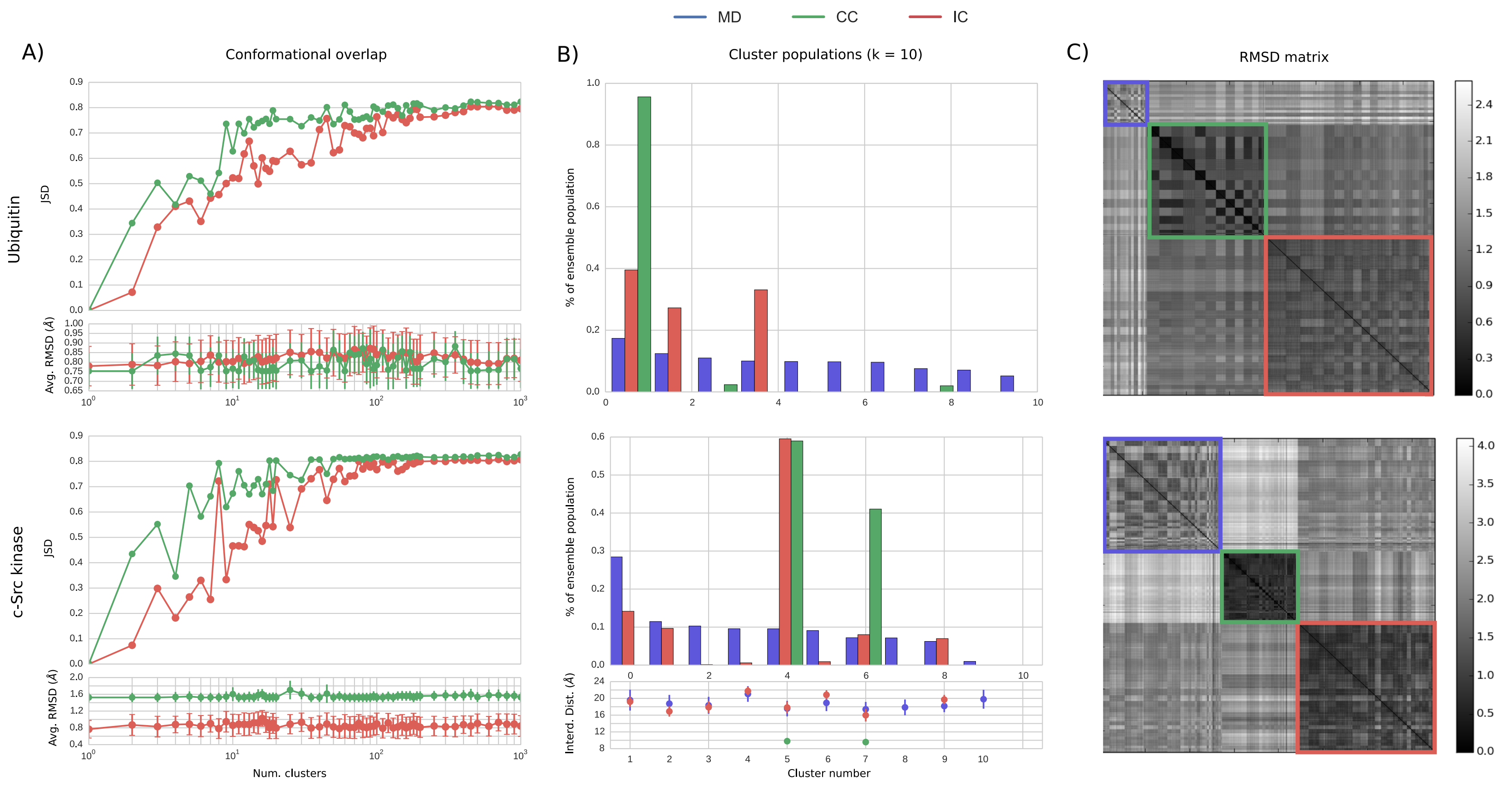

Figure 7: A) JSD of the NMA methods and MD for a different number of clusters ($1 \leq k \leq 1000$). B) Detail of the distributions per cluster for $k = 10$. C) C_α RMSD matrices for the whole ensemble of simulations. The submatrices of MD and NMA simulations have been highlighted.

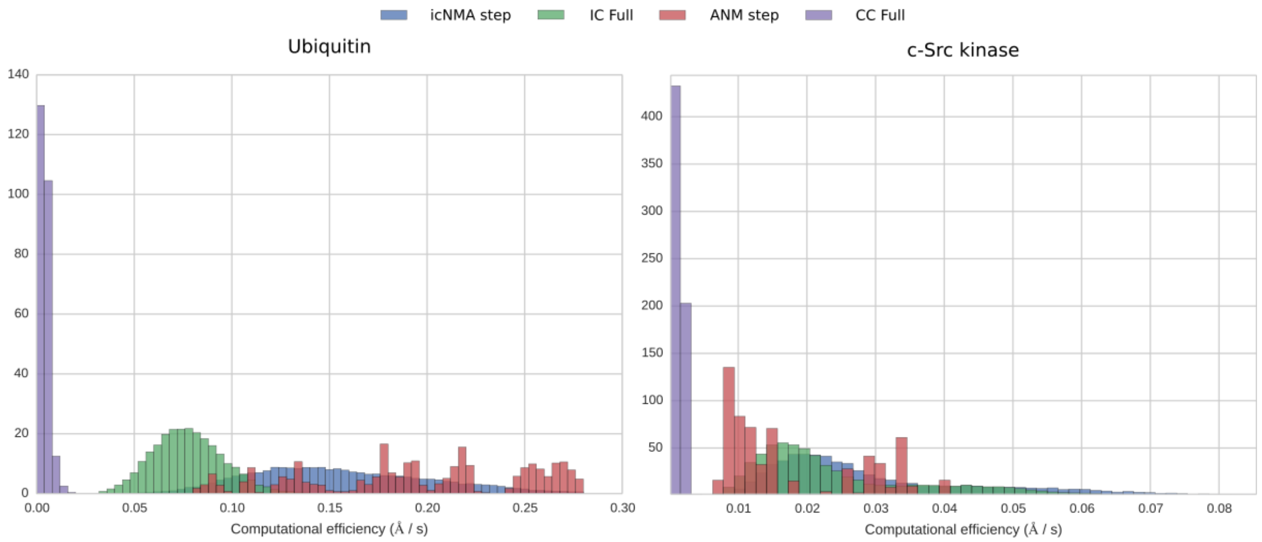

Figure 8: Distribution of the computational efficiency for both methods and systems. Both NMA steps (red and blue) look to be able to perform similarly. However, when comparing with PELE full iteration (purple), it can be clearly seen that the IC-based method has a larger efficiency, even when the amortized side chain perturbation time is considered (green).

Table 4: Average time values of an icNMA iteration, the side chain perturbation step, the ANM step, and a full MC iteration of PELE. The need of the relaxation phase in the CC-based PELE algorithm makes the overall time of an iteration clearly slower than an icNMA iteration. The simulations were run in AMD Opteron 6238 @ 2.60GHz nodes with 4Gb of RAM per node. Standard deviations are included between parentheses.

System	icNMA step (s)	Side. Perturb. All steps (s)	ccNMA step (s)	PELE iteration (s)
Ubiquitin	2.945 (1.456)	21.327 (2.503)	1.376 (0.704)	20.065 (5.624)
c-Src kinase	11.648 (3.273)	27.917 (2.958)	6.106 (0.679)	57.9 (9.233)

4 Conclusions

There is a high interest in developing faster sampling techniques for modeling backbone flexibility in proteins, with normal mode approximations such as ANM becoming a popular alternative. The modes obtained using ANM, however, describe the movement of only one atom per residue (the C_α atom) and applying this movement to the remaining atoms is not trivial. In PELE, this is achieved by applying a minimization. However, in this step the covalent topology of the protein is often unphysically distorted, which requires the introduction of a relaxation phase where a system-wide minimization is performed.

In this article we have presented a new MC method that handles the protein backbone changes using IC NMA. The application of the internal modes through a geometrical manipulation of torsions does not distort the covalent topology, allowing us to to generate more energetically favorable conformation proposals than our previous method. Another fundamental advantage of the use of torsional modes is their increased collectivity and robustness, which alleviates the "tip effect" problem, facilitating the fast traversal of the conformational space.

The elimination of the ANM minimization and the unneeded relaxation phase has two major consequences. First, as both the side chain prediction step and the system-wide constrained minimization are computationally costly processes, the new method shows a great speedup (~5-7x) without decreasing the RMSD of the perturbations. Second, as these minimizations amplify the solvent model bias towards compact structures, the new method shows SASA and radius of gyration distributions which are closer to those observed in MD. Moreover, our analyses have also shown improvements in the exploration of the conformational space: the icNMA algorithm is able to explore similar regions to MD and generate structures with closer RMSD than the CC-based method.

The main drawback we have found is that, although the icNMA method seems to produce relative fluctuations that are in better accordance with MD than the ANM-based method, these fluctuations are smaller and show a lower baseline than MD. This is the consequence of one of the main limitations of applying NMA-based techniques, where local flexibility is not well represented by higher frequency modes.

Overall, implementing icNMA results in better agreement with MD explicit solvent simulations. Using internal coordinates seems to be a promising technique for speeding up the induced fit studies in PELE and supposes a step forward in terms of the quality of the flexibility handling.

Acknowledgement

The authors thank D. E. Shaw Research lab. for providing the kinase MD coordinates and Dr. López Blanco for sharing the code developed in his thesis. This work was supported by the CTQ-48287-R projects of the Spanish Ministry of Economy and Competitiveness (MINECO) and the grant SEV-2011-00067 of Severo Ochoa Program, awarded by the Spanish Government.

References

(1) Erickson, J. A.; Jalaie, M.; Robertson, D. H.; Lewis, R. A.; Vieth, M. *J. Med. Chem. 47*, 45–55.

(2) Huang, S.-Y.; Zou, X. *Proteins 66*, 399–421.

(3) Totrov, M.; Abagyan, R. *18*, 178–184.

(4) Sherman, W.; Day, T.; Jacobson, M. P.; Friesner, R. A.; Farid, R. *J. Med. Chem. 49*, 534–553.

(5) Legge, F. S.; Budi, A.; Treutlein, H.; Yarovsky, I. *Biophys. Chem. 119*, 146–157.

(6) Jorgensen, W. L.; Tirado-Rives, J. *J. Comput. Chem. 26*, 1689–1700.

(7) Borrelli, K. W.; Vitalis, A.; Alcantara, R.; Guallar, V. *J. Chem. Theory Comput. 1*, 1304–1311.

(8) Shan, Y.; Kim, E. T.; Eastwood, M. P.; Dror, R. O.; Seeliger, M. A.; Shaw, D. E. *J. Am. Chem. Soc. 133*, 9181–9183.

(9) Kresten Lindorff-Larsen1, R. O. D., Stefano Piana; Shaw, D. E. *334*, 517–520.

(10) De Vivo, M.; Masetti, M.; Bottegoni, G.; Cavalli, A. *J. Med. Chem.*

(11) Cole, D. J.; Tirado-Rives, J.; Jorgensen, W. L. *Biochim. Biophys. Acta, Gen. Subj 1850*, 966–971.

(12) Frenkel, D.; Smit, B. *Understanding molecular simulation: from algorithms to applications*, 2nd ed.; Computational science series 1; Academic Press.

(13) Rapaport, D. C.; Scheraga, H. A. *Macromolecules 14*, 1238–1246.

(14) Paine, G. H.; Scheraga, H. A. *Biopolymers 24*, 1391–1436.

(15) Lotan, I.; Schwarzer, F.; Latombe, J.-C. In *Algorithms in Bioinformatics*; Benson, G., Page, R. D. M., Eds.; Lecture Notes in Computer Science 2812; Springer Berlin Heidelberg, pp 354–373.

(16) Tang, K.; Zhang, J.; Liang, J. *PLoS Comput. Biol. 10*.

(17) Esteban-Martín, S.; Fenwick, R. B.; Ådén, J.; Cossins, B.; Bertoncini, C. W.; Guallar, V.; Wolf-Watz, M.; Salvatella, X. *PLoS Comput. Biol. 10*, e1003721.

(18) Hosseini, A.; Espona-Fiedler, M.; Soto-Cerrato, V.; Quesada, R.; Pérez-Tomás, R.; Guallar, V. *PLoS ONE 8*, e57562.

(19) Eyer, L.; Valdés, J. J.; Gil, V. A.; Nencka, R.; Hřebabecký, H.; Šála, M.; Salát, J.; Černý, J.; Palus, M.; Clercq, E. D.; Růžek, D. *Antimicrob. Agents Chemother. 59*, 5483–5493.

(20) Grebner, C.; Iegre, J.; Ulander, J.; Edman, K.; Hogner, A.; Tyrchan, C. *J. Chem. Inf. Model.*

(21) Monza, E.; Lucas, M. F.; Camarero, S.; Alejaldre, L. C.; Martínez, A. T.; Guallar, V. *J. Phys. Chem. Lett. 6*, 1447–1453.

(22) Acebes, S.; Fernandez-Fueyo, E.; Monza, E.; Lucas, M. F.; Almendral, D.; Ruiz-Dueñas, F. J.; Lund, H.; Martinez, A. T.; Guallar, V. *ACS Catal. 6*, 1624–1629.

(23) Takahashi, R.; Gil, V. A.; Guallar, V. *J. Chem. Theory Comput. 10*, 282–288.

(24) Atilgan, A. R.; Durell, S. R.; Jernigan, R. L.; Demirel, M. C.; Keskin, O.; Bahar, I. *Biophys. J. 80*, 505–515.

(25) Rueda, M.; Chacón, P.; Orozco, M. *Structure 15*, 565–575.

(26) Ahmed, A.; Villinger, S.; Gohlke, H. *Proteins 78*, 3341–3352.

(27) Kitao, A.; Hirata, F.; Gō, N. *158*, 447–472.

(28) Skjaerven, L.; Martinez, A.; Reuter, N. *Proteins 79*, 232–243.

(29) Northrup, S. H.; McCammon, J. A. *Biopolymers 19*, 1001–1016.

(30) Li, Z.; Scheraga, H. A. *Proc. Natl. Acad. Sci. U.S.A. 84*, 6611–6615.

(31) Jorgensen, W. L.; Tirado-Rives, J. *J. Phys. Chem. 100*, 14508–14513.

(32) Kidera, A. *Int. J. Quantum Chem 75*, 207–214.

(33) Noguti, T.; Gō, N. *Biopolymers 24*, 527–546.

(34) Lopez-Blanco, J. R.; Garzón, J. I.; Chacón, P. *Bioinformatics 27*, 2843–2850.

(35) Kovacs, J. A.; Cavasotto, C. N.; Abagyan, R. *J. Comput. Theor. Nanosci. 2*, 354–361.

(36) Eckart, C. *Phys. Rev. A 47*, 552–558.

(37) Noguti, T.; Gō, N. *J. Phys. Soc. Jpn. 52*, 3283–3288.

(38) Abe, H.; Braun, W.; Noguti, T.; Gō, N. *Comput. Chem. 8*, 239–247.

(39) Choi, V. *J. Chem. Inf. Model. 46*, 438–444.

(40) Tamar Schlick, A. L. F. *18*, 71–111.

(41) Genheden, S.; Ryde, U. *Phys. Chem. Chem. Phys. 14*, 8662–8677.

(42) Henzler-Wildman, K.; Kern, D. *Nature 450*, 964–972.

(43) Plattner, N.; Noé, F. *Nat. Commun. 6*, 7653.

(44) Glickman, M. H.; Ciechanover, A. *82*, 373–428.

(45) Schnell, J. D.; Hicke, L. *J. Biol. Chem. 278*, 35857–35860.

(46) Mukhopadhyay, D.; Riezman, H. *315*, 201–205.

(47) Vijay-Kumar, S.; Bugg, C. E.; Cook, W. J. *J. Mol. Biol. 194*, 531–544.

(48) Bang, D.; Makhatadze, G. I.; Tereshko, V.; Kossiakoff, A. A.; Kent, S. B. *Angew. Chem. Int. Ed. 44*, 3852–3856.

(49) Liu, G.; Forouhar, F.; Eletsky, A.; Atreya, H. S.; Aramini, J. M.; Xiao, R.; Huang, Y. J.; Abashidze, M.; Seetharaman, J.; Liu, J.; Rost, B.; Acton, T.; Montelione, G. T.; Hunt, J. F.; Szyperski, T. *J. Struct. Funct. Genomics 10*, 127–136.

(50) Alexeev, D.; Barlow, P. N.; Bury, S. M.; Charrier, J.-D.; Cooper, A.; Hadfield, D.; Jamieson, C.; Kelly, S. M.; Layfield, R.; Mayer, R. J.; McSparron, H.; Price, N. C.; Ramage, R.; Sawyer, L.; Starkmann, B. A.; Uhrin, D.; Wilken, J.; Young, D. W. *ChemBioChem 4*, 894–896.

(51) Levin-Kravets, O.; Shohat, N.; Prag, G. *Biochemistry 54*, 4704–4710.

(52) Fasshuber, H. K.; Lakomek, N.-A.; Habenstein, B.; Loquet, A.; Shi, C.; Giller, K.; Wolff, S.; Becker, S.; Lange, A. *Prot. Sci. 24*, 592–598.

(53) Kony, D. B.; Hünenberger, P. H.; van Gunsteren, W. F. *Prot. Sci. 16*, 1101–1118.

(54) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. *Proc. Natl. Acad. Sci. U.S.A. 110*, 5915–5920.

(55) Ganoth, A.; Tsfadia, Y.; Wiener, R. *J. Mol. Graphics Modell. 46*, 29–40.

(56) Beauchamp, K. A.; Lin, Y.-S.; Das, R.; Pande, V. S. *J. Chem. Theory Comput. 8*, 1409–1414.

(57) Leherte, L.; Vercauteren, D. P. *Sci. China Chem. 57*, 1340–1354.

(58) Raval, A.; Piana, S.; Eastwood, M. P.; Shaw, D. E. *Prot. Sci. 25*, 19–29.

(59) Cossins, B. P.; Hosseini, A.; Guallar, V. *J. Chem. Theory Comput. 8*, 959–965.

(60) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. *J. Comput. Chem. 26*, 1701–1718.

(61) Jorgensen, W. L.; Tirado-Rives, J. *J. Am. Chem. Soc. 110*, 1657–1666.

(62) Aleshin, A.; Finn, R. S. *Neoplasia 12*, 599–607.

(63) Schindler, T.; Sicheri, F.; Pico, A.; Gazit, A.; Levitzki, A.; Kuriyan, J. *Mol. Cell 3*, 639–648.

(64) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. *Proteins 65*, 712–725.

(65) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. *Proteins 78*, 1950–1958.

(66) Bowers, K. J.; Chow, D. E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossvary, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; Shaw, D. E. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. Proceedings of the ACM/IEEE SC 2006 Conference. pp 43–43.

(67) Shaw, D. E.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Lerardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Deneroff, M. M.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J. *51*, 91.

(68) Boehr, D. D.; Nussinov, R.; Wright, P. E. *Nat. Chem. Biol. 5*, 789–796.

(69) Csermely, P.; Palotai, R.; Nussinov, R. *Trends Biochem. Sci 35*, 539–546.

(70) Moret, M. A.; Santana, M. C.; Zebende, G. F.; Pascutti, P. G. *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys. 80*, 041908.

(71) Lobanov, M. Y.; Bogatyreva, N. S.; Galzitskaya, O. V. *Mol. Biol. 42*, 623–628.

(72) Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graph. Model. 14*, 33–38.

(73) Lyman, E.; Zuckerman, D. M. *Biophys. J. 91*, 164–172.

(74) Lindorff-Larsen, K.; Ferkinghoff-Borg, J. *PLoS ONE 4*, e4203.

(75) Gil, V. A.; Guallar, V. *J. Chem. Theory Comput. 10*, 3236–3243.

(76) Endres, D. M.; Schindelin, J. E. *49*, 1858–1860.

(77) Zheng, W. *Biophys. J. 98*, 3025–3034.

(78) Gelman, A.; Roberts, G.; Gilks, W. *Bayesian statistics, 5 (Alicante, 1994)*; Oxford Univ. Press, pp 599–607.

(79) Onufriev, A.; Bashford, D.; Case, D. A. *Proteins 55*, 383–394.

(80) Zhang, W.; Ganguly, D.; Chen, J. *PLoS Comput. Biol. 8*, e1002353.

(81) Foloppe, N.; Chen, I.-J. *Bioorg. Med. Chem.*

(82) Jaramillo, A.; Wodak, S. J. *Biophys. J. 88*, 156–171.

(83) Formaneck, M.; Cui, Q. *J. Comput. Chem. 27*, 1923.

(84) Chen, J.; Iii, C. L. B. *Phys. Chem. Chem. Phys. 10*, 471–481.

(85) Stultz, C. M. *J. Phys. Chem. B 108*, 16525–16532.

(86) Boggon, T. J.; Eck, M. J. *Oncogene 23*, 7918–7927.

(87) Hayward, S.; Kitao, A.; Gō, N. *Prot. Sci. 3*, 936–943.