SBMOpenMM: A builder of Structure-Based Models for OpenMM

Martin Floor,[a,b,#] Kengjie Li,[d] Miquel Estévez-Gay,[e] Luis Agulló,[c] Pau Marc Muñoz,[a] Jenn K. Hwang,[d] Sílvia Osuna,[e,f] Jordi Villà-Freixa[a,b]

[a]Department of Basic Sciences, Faculty of Medicine and Health Sciences, Universitat Internacional de Catalunya, 08195 Sant Cugat del Vallès, Spain
[b]Department of Biosciences, Faculty of Sciences and Technology, Universitat de Vic - Universitat Central de Catalunya, 08500 Vic, Spain
[c]Faculty of Medicine, Universitat de Vic - Universitat Central de Catalunya, Vic, Spain
[d]Warshel Institute of Computational Biology, The Chinese University of Hong Kong, Shenzhen, China
[e]CompBioLab group, Institut de Química Computacional i Catàlisi (IQCC) and Departament de Química, Universitat de Girona, Girona, Spain
[f]ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain

[#]corresponding author mfloor@uic.es

Keywords:
Structure-based molecular dynamics, protein folding, OpenMM, SBMOpenMM, Markov State Models

Abstract

Molecular dynamics (MD) simulations have become a standard tool to correlate the structure and function of biomolecules, and significant advances have been made in the study of proteins and their complexes. Despite its, in principle, correct behavior given a proper force field is used, a major drawback of MD simulations is the difficulty and cost of obtaining converged results, especially when exploring a potential energy surface containing considerable energy barriers. This prevents the wide use of MD calculations to determine the thermodynamic properties of biomolecular processes. Indeed, this is true when one considers the conformational entropy of such processes, which is ultimately a critical component in assessing the simulations' convergence. Alternatively, a wide range of Structure-Based Models (SBMs) has been used in the literature to unravel the basic mechanisms of biomolecular dynamics. SBMs introduce simplifications that focus on the relevant aspects of the physical process under study. The main limitations of such SBMs include the necessity to tailor-made models targeting particular biophysical questions and to adapt them for efficient computational frameworks, using advanced software-hardware interfaces (e.g., GPU-based simulations). Here we introduce SBMOpenMM, a python library to build SBMs, that uses the OpenMM framework to create and run SBM simulations. The code is flexible, user-friendly, and profits from high customizability and GPU performance provided by the OpenMM platform. We demonstrate its use in the evaluation of the two-step folding process of FoxP1 transcription factor protein. Our results indicate that the newly developed SBM can be successfully applied to elucidating the underlying mechanisms of biomolecular processes.

Background

Proteins are structurally fluctuating macromolecular systems that perform the vast majority of functions of biological cells. Protein dynamics have different timescale manifestations related to various physical events, ranging from fast local flexibility to slower collective motions [1]. Protein atoms' movements ultimately govern the kinetics and thermodynamics of folding, binding, catalysis, among other biochemical activities [2–4].

Molecular dynamics (MD) has become an essential tool for the computational study of biological polymers. Detailed physical observations can be directly obtained otherwise not directly accessible by traditional wet-lab experimental studies. Nonetheless, the computational cost for running MD simulations in the timescales of protein folding or other functions still hinders its application for many relevant processes with significant activation barriers. Many efforts have been devoted to enhancing conformational space sampling by improving searching algorithms [5], simplified force fields [6–8], and accelerated algorithm execution through code optimization and parallelization [9].

Among several MD techniques, dating back to the pioneering work of Warshel and Levitt [10,11], Structure-Based Models (SBMs) emerged as a simplified methodology for studying protein folding dynamics [12,13]. SBMs are based on simplifying assumptions, rooted in energy landscape theory for protein folding, that capture essential physical aspects of natural proteins and bridge the gap between physical complexity and computational efficiency. These simplifications allow us to obtain a reasonable kinetic and equilibrium characterization of protein systems to be studied within the framework of statistical mechanics.

SBMs idealize the energy landscape based on two main ideas. First, natural proteins have evolved to avoid kinetic traps along the folding coordinate; this prevents the formation of long-lived folding intermediaries containing non-native interactions that could frustrate the folding reaction [14]. Secondly, protein evolution has maximized the energy gap between the natively folded state and other competing configurations; these focused simplifications increase the number of protein molecules that populate the native basin and remove the excess of kinetic frustration. SBMs exploit these facts by focusing on the native structure's geometry and contact information, exploring a conformational landscape based entirely on the native topology's constraints. In this way, one can separate energetic from topological contributions, showing that essential features of protein folding are mainly encoded by the organization of the folded state [15]. Following the same concept, SBM simulations have also been used to study protein-protein binding mechanisms or ligand-induced conformational changes.[16] Additionally, multi-basin SBM potentials have served to study other complex mechanisms in protein dynamics, such as folding to competing native configurations, conformational shifts, domain swapping, knotting, or drastic structural rearrangements between functional and dysfunctional conformations. For an extensive review on SBM applications, see [16].

Current implementations of SBM software packages lack the flexibility to manipulate force field terms and parameters easily. Additionally, in some cases, the parallelization does not support GPU acceleration because they are encoded in outdated versions of MD software or have force field definitions that do not support it [17,18]. Because SBM is an ongoing practical and meaningful research methodology, it would greatly benefit from a fast, accessible, flexible, and easy-to-set-up implementation that facilitates simulation and force field experimentation. To this end, we have developed SBMOpenMM, a python library, to set-up SBMs using the OpenMM toolkit framework for MD simulations [19]. The python code is flexible, user-friendly, and profits from the high customizability and performance provided by the OpenMM platform. We hope this will foster further experimentation in the field in an open-source collaborative manner.

This study demonstrates the code's usage in exploring the two-state folding process of the forkhead box P1 transcription factor protein (FoxP1).

Implementation

SBMOpenMM uses two files as input; a Protein Data Bank (PDB) format file for the protein structure and a contact file for specifying the native contact interactions. The set of bonded interactions are defined from the PDB structure's connectivity. These include definitions of atoms participating in bonds, angles, and proper, improper, and planar torsions, together with their equilibrium distances. Likewise, nonbonded interactions are defined from the atomic pairs listed in the contact file, and their equilibrium distances are calculated from the input structure. This connectivity information, together with the set of force constant parameters, are used by the library to construct the relevant forces and system OpenMM objects, directly employed by the OpenMM MD engine to start a simulation.

Library data structure and utilization

The SBMOpenMM library contains methods to define the SBM forcefield's force composition and set default or custom parameters. To this end, the python library is composed of three major classes (Figure S2):

- ➔ *`geometry`*
- ➔ *`system`*
- ➔ *`models`*

The first class, *`geometry`*, contains methods for calculating the input structures' geometrical parameters, including equilibrium distances of contacts, bonds, angles, and torsional degrees of freedom.

The library's main class is the *`system`* class, which contains all the SBM system definitions. These definitions include the connectivity, initial coordinates, force field parameters, and the OpenMM forces objects to be included. Several attributes and methods inside this class allow customizing the forces and parameters in the SBM forcefield definition before creating an OpenMM system object.

Finally, the *`models`* class automatizes the generation of default SBMOpenMM *`system`*-class objects containing the default forcefield parameters to create coarse-grained or all-atom SBMs, or their corresponding multi-basin versions, ready to be simulated with the OpenMM engine.

A simple python code to run a default all-atom SBM simulation with OpenMM is:

```
import sbmOpenMM
from simtk.openmm.app import *
from simtk.openmm import *
from simtk.unit import *

sbmAA = sbmOpenMM.models.getAllAtomModel(pdb_file, contact_file)

integrator = LangevinIntegrator(temperature*kelvin, 1.0/picosecond,
0.002*picoseconds)
simulation = Simulation(sbmAA.topology, sbmAA.system, integrator)
simulation.context.setPositions(sbmAA.positions)
simulation.reporters.append(DCDReporter('AAModel_traj.dcd', 10000))
sbmReporter  =  sbmOpenMM.reporter.sbmReporter('AAModel_energy.data',
                                                100,           step=True,
                                                potentialEnergy=True,
                                                temperature=True,
                                                sbmObject=sbmAA)

simulation.reporters.append(sbmReporter)

simulation.step(n_steps)
```

Here, the *sbmAA* object, created with SBMOpenMM, contains the SBM "topology," "system," and "positions" attributes to be passed directly to the OpenMM library. After the *sbmAA* object is created, the usual steps for running simulations with OpenMM are followed. First, an OpenMM *integrator* object is created and loaded into an OpenMM *simulation* object. Here, we pass the *topology,* and OpenMM *system* attributes contained in the *sbmAA* object. Then, the initial positions in the *sbmAA* object are passed to the simulation's OpenMM context attribute. To write simulation data into files, we define two reporter classes; a trajectory reporter (an OpenMM *DCDReporter* class) and a state data reporter (StateDataReporter class) for storing energies and other simulation parameters. The sbmReporter is an inherited class of the OpenMM StateDataReporter that allows communicating the SBM energies if an SBM object (here *sbmAA*) is passed to it with the sbmObject option. Finally, the above code's last line will advance the simulation for *n_steps* number of steps, thus starting the MD simulation.

Similarly, if the system should be simulated using instead a coarse-grained alpha-carbon (CA) SBM, creating the SBM system class changes to:

```
sbmCA = sbmOpenMM.models.getCAModel(pdb_file, contact_file)
```

Here, sbmCA is analogous to the above *sbmAA* object but contains the definitions for a default coarse-grained CA SBM.

The general workflow for running a SBM simulation with SBMOpenMM is shown in Figure 1.

To aid in the visualization of trajectories is useful to generate a PDB format file containing only the atoms in the SBM system. There is a dumpStructure() method inside the SBM system class objects, which can be called to write a PDB file containing only the simulated atoms:

```
sbmCA.dumpStructure('ca_output_file.pdb')
```

To aid in the tracking or manual modification of parameters, there is a `dumpForceFieldData` and `loadForcefieldFromFile` methods inside the SBM `system` class to write or load the SBM forcefield definition to or from a file. These methods can also be useful when importing other SBMs into this library.

For further examples and additional information on creating and customizing SBM force fields with SBMOpenMM, please refer to the supplementary information and the hosting web site's documentation and tutorials.

https://github.com/CompBiochBiophLab/sbm-openmm

Example of use

To test and portray the SBMOpenMM library's utility, we ran folding simulations for the DNA binding domain of the forkhead box P1 transcription factor protein (FoxP1). FoxP1 is a small globular protein necessary for the development of various organs in mammals [20]. Intriguingly, FoxP1 generates dimeric structures through a domain swapping mechanism relevant to its biological function as a transcription factor [21,22].

Employing the monomeric structure of the FoxP1 DNA-binding domain (Figure 3A) [22], we carried out fifteen 10 μs folding/unfolding equilibrium replicas using an all-atom SBM at the folding temperature of the system (i.e., the temperature that maximizes folding and unfolding transitions). The native contacts employed to define the SBM were calculated using the shadow contact algorithm [23]. The code for running the simulations was essentially the same as the one presented in the utilization example in the implementation section. The simulations show different numbers of folding/unfolding transitions, ranging from one to eight, indicating independence on the course of phase space sampled among them (Figure 2 and S1).

The time needed to run this all-atom SBM (747 atoms) simulation is 224 times faster than the time needed for its fully-solvated version (1.2-nanometer solvent buffer simulation box, employing an unfolded FoxP1 conformation; 10,973 atoms) using the same hardware (GeForce GTX 1050 mobile).

We analyzed the simulated trajectories using the Markov-State Model (MSM) framework implemented in the pyEMMA package [24] (for details on MSM construction and validation, see the supplementary information). The system's free energy was projected into the two slowest Time-lagged Independent Component Analysis (TICA) dimensions [25] (Figure 3B). The simulation reproduces the experimental folding mechanism of FoxP1 as a two-state folder [21].

To follow the unfolding mechanism's progression, we estimated the probability of contact formation at the folded, TS region, and unfolded configurations (Figure 3C). While the folded configuration retains most of the native contacts, it shows diminished interactions for the contacts made by the N- and C-terminal residues (Figure 3C, left matrix “b” triangle). Most native contacts are already lost at the TS; however, contacts made between strands 1, 2, and 3 hold the structure and are the last to be broken (Figure 3C, right matrix “c” triangle).

Discussion

Due to the simplifications made over the force field definition, SBM simulations converge extraordinarily fast compared to standard MD, allowing, for example, simulating the folding of a small protein in equilibrium conditions by using only modern personal computer hardware, but also allowing to explore other biomolecular processes involving considerable energy barriers. Even though SBMs simplifications can overlook relevant pathways that include non-native interactions or explicit solvent definitions, an idealized study can serve many purposes before moving to more complex, and definitely more expensive, representations. On the one hand, SBMs are used to understand the topological restrictions that a specific fold imposes on the protein's dynamical behavior. By later, including different or additional forcefield terms, deviations from this idealized model can be attributed to the specific physical parameters incorporated in the simulation. On the other hand, SBM can serve as an initial step before moving into simulations employing state-of-the-art MD force fields following adaptive sampling techniques to study complex biomolecular phenomena [26]. In such approaches, relevant conformations, carefully extracted from the simplified simulation, can be used as seeds to optimize the phase-space sampling by more costly simulations, making SBM a useful and practical tool to conduct theoretical research in protein biophysics.

A protein system can have more than one relevant native configuration, which can be assessed using multi-basin potential energy models. Their representation can be achieved either macroscopically by coupling single-minimum Hamiltonians or microscopically by explicitly adding multiple energy minima at different distances for shared contacts between the different configurations. To this end, SBMOpenMM also contains multi-basin SBM versions based on the default AA and CA SBM models. Furthermore, to expand the applicability of SBM, many other modifications to the Hamiltonian can be explored. These include, but are not limited to, adding electrostatic forces or explicit hydrogen representations, modifying contact strengths or torsional contributions, considering idiosyncrasies of the amino acid or atom compositions. All these modifications allow us to ascertain and examine their specific contributions to protein dynamics phenomena.

In summary, SBMOpenMM paves the way to explore complex structure-function relationships in biomolecular processes, including protein folding, allosterism, domain swapping, protein-ligand binding effects, or protein-protein interactions, among others.

Figures

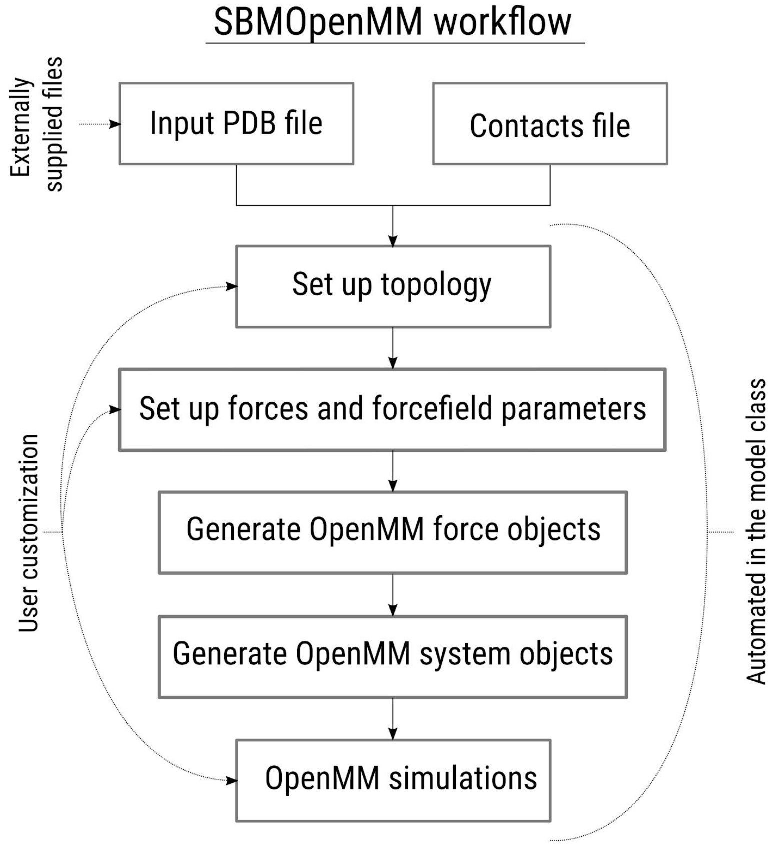
SBMOpenMM workflow
Externally supplied files
Input PDB file
Contacts file
Set up topology
Set up forces and forcefield parameters
Generate OpenMM force objects
Generate OpenMM system objects
OpenMM simulations
User customization
Automated in the model class

Figure 1. Workflow to set up a simulation with SBMOpenMM. Only a PDB and a contact file are needed for running SBM simulations. The library contains automated methods to set up the geometric and force field parameters and generate the force objects that will act during the simulation. When all the parameters and forces are ready, SBMOpenMM generates an OpenMM system class object for running MD simulations. Most of these steps can also be user-customized; however, to make a faster deployment of models, default SBMs can be called directly from the *model* class.

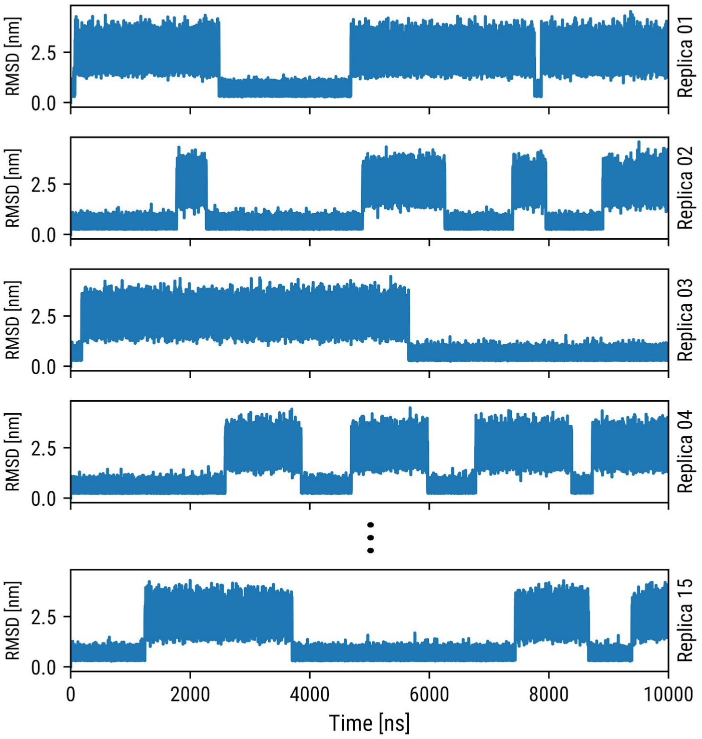

Figure 2. FoxP SBM folding simulations. Five of the fifteen 10 μs SBM replicas at the folding temperature of the SBM system. Each plot shows the evolution of the RMSD to the native structure in nanometers. High and low RMSD values represent the time spent by the trajectory at the unfolded and folded configurations, respectively. The plot for all replicas ran in this study is shown in Figure S1.

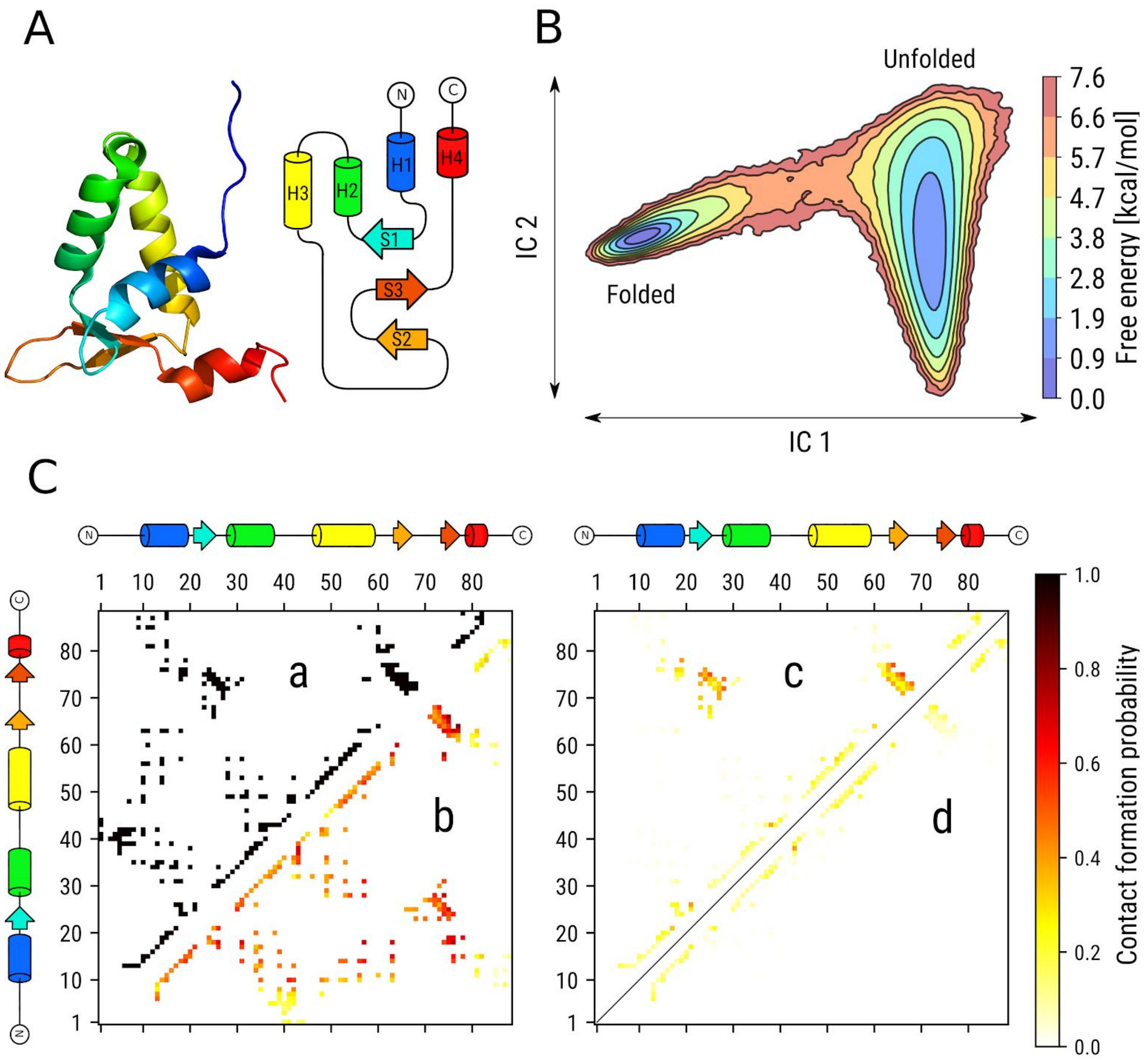

Figure 3. Analysis of the FoxP1 SBM folding simulation. (A) Tertiary structure and topological arrangement of the FoxP1 domain. Loops in this topology diagram are not in scale with their sequence length. (B) Free energy profile projected into the two slowest time-lagged independent component analysis (TICA) coordinates (IC) based on native contact distances. FoxP1 folds in a two-state mechanism, clearly visible as two characteristic minima; the folded state (left) and the unfolded state (right). (C) FoxP1 tertiary structure per-residue contact map (a); and contact formation probability at the folded state (b), transition state (c), and unfolded state (d) for the SBM folding simulation.

Availability

Project name: SBMOpenMM
Project home page: https://github.com/CompBiochBiophLab/sbm-openmm
Operating system(s): Platform independent
Programming language: Python3

Other requirements: OpenMM, numpy
License: MIT

Author Contributions

MF and KJL developed the SBMOpenMM package. MF, MEG, and SO carried out the MSM analysis. LA, JKH, and JVF provided planning and guidance for the project. PM provided testing of the package. All authors contributed to write, read, and approve the final version of the manuscript. MF was a major contributor in writing the manuscript.

Corresponding author

Correspondence to Martin Floor (mfloor@uic.es).

Competing interests

The authors declare that they have no competing interests.

Funding

A UVic-UCC Ph.D. fellowship funded the work by MF. JVF, MF, LA, and PM thank Generalitat de Catalunya for the consolidated group BMS (2017 SGR-901). The SBMOpenMM software has been developed within the BIO2017-83650-P project, funded by the Spanish Subprograma Estatal de Generación del Conocimiento. MEG and SO thank Generalitat de Catalunya for the emerging group CompBioLab (2017 SGR-1707) and Spanish MINECO for project PGC2018-102192-B-I00. SO is grateful for the European Research Council (ERC) funding under the European Union's Horizon 2020 research and innovation program (ERC-2015-StG-679001).

References

(1) Berendsen, H. Collective Protein Dynamics in Relation to Function. *Current Opinion in Structural Biology*. 2000, pp 165–169. https://doi.org/10.1016/s0959-440x(00)00061-0.

(2) Kern, D.; Zuiderweg, E. R. P. The Role of Dynamics in Allosteric Regulation. *Current Opinion in Structural Biology*. 2003, pp 748–757. https://doi.org/10.1016/j.sbi.2003.10.008.

(3) Grant, B. J.; Gorfe, A. A.; McCammon, J. A. Large Conformational Changes in Proteins: Signaling and Other Functions. *Curr. Opin. Struct. Biol.* **2010**, *20* (2), 142–147.

(4) Villà, J.; Warshel, A. Energetics and Dynamics of Enzymatic Reactions. *The Journal of Physical Chemistry B*. 2001, pp 7887–7907. https://doi.org/10.1021/jp011048h.

(5) Yang, Y. I.; Shao, Q.; Zhang, J.; Yang, L.; Gao, Y. Q. Enhanced Sampling in Molecular Dynamics. *J. Chem. Phys.* **2019**, *151* (7), 070902.

(6) Barnoud, J.; Monticelli, L. Coarse-Grained Force Fields for Molecular Simulations. *Methods in Molecular Biology*. 2015, pp 125–149.

https://doi.org/10.1007/978-1-4939-1465-4_7.

(7) Poma, A. B.; Cieplak, M.; Theodorakis, P. E. Combining the MARTINI and Structure-Based Coarse-Grained Approaches for the Molecular Dynamics Studies of Conformational Transitions in Proteins. *J. Chem. Theory Comput.* **2017**, *13* (3), 1366–1374.

(8) Sutto, L.; Mereu, I.; Gervasio, F. L. A Hybrid All-Atom Structure-Based Model for Protein Folding and Large Scale Conformational Transitions. *J. Chem. Theory Comput.* **2011**, *7* (12), 4208–4217.

(9) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. GPU-Accelerated Molecular Modeling Coming of Age. *Journal of Molecular Graphics and Modelling*. 2010, pp 116–125. https://doi.org/10.1016/j.jmgm.2010.06.010.

(10) Levitt, M.; Warshel, A. Computer Simulation of Protein Folding. *Nature*. 1975, pp 694–698. https://doi.org/10.1038/253694a0.

(11) Fan, Z. Z.; Hwang, J.-K.; Warshel, A. Using Simplified Protein Representation as a Reference Potential for All-Atom Calculations of Folding Free Energy. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*. 1999, pp 77–80. https://doi.org/10.1007/s002140050516.

(12) Taketomi, H.; Ueda, Y.; Gō, N. STUDIES ON PROTEIN FOLDING, UNFOLDING AND FLUCTUATIONS BY COMPUTER SIMULATION: I. The Effect of Specific Amino Acid Sequence Represented by Specific Inter-Unit Interactions. *Int. J. Pept. Protein Res.* **1975**, *7* (6), 445–459.

(13) Nymeyer, H.; García, A. E.; Onuchic, J. N. Folding Funnels and Frustration in off-Lattice Minimalist Protein Landscapes. *Proc. Natl. Acad. Sci. U. S. A.* **1998**, *95* (11), 5921–5928.

(14) Dill, K. A.; Chan, H. S. From Levinthal to Pathways to Funnels. *Nat. Struct. Biol.* **1997**, *4* (1), 10–19.

(15) Clementi, C.; Nymeyer, H.; Onuchic, J. N. Topological and Energetic Factors: What Determines the Structural Details of the Transition State Ensemble and "En-Route" Intermediates for Protein Folding? An Investigation for Small Globular Proteins. *J. Mol. Biol.* **2000**, *298* (5), 937–953.

(16) Noel, J. K.; Onuchic, J. N. The Many Faces of Structure-Based Potentials: From Protein Folding Landscapes to Structural Characterization of Complex Biomolecules. *Computational Modeling of Biological Systems*. 2012, pp 31–54. https://doi.org/10.1007/978-1-4614-2146-7_2.

(17) Noel, J. K.; Levi, M.; Raghunathan, M.; Lammert, H.; Hayes, R. L.; Onuchic, J. N.; Whitford, P. C. SMOG 2: A Versatile Software Package for Generating Structure-Based Models. *PLoS Comput. Biol.* **2016**, *12* (3), e1004794.

(18) Lutz, B.; Sinner, C.; Bozic, S.; Kondov, I.; Schug, A. Native Structure-Based Modeling and Simulation of Biomolecular Systems per Mouse Click. *BMC Bioinformatics* **2014**, *15*, 292.

(19) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLoS Comput. Biol.* **2017**, *13* (7), e1005659.

(20) Gabut, M.; Samavarchi-Tehrani, P.; Wang, X.; Slobodeniuc, V.; O'Hanlon, D.; Sung, H.-K.; Alvarez, M.; Talukder, S.; Pan, Q.; Mazzoni, E. O.; Nedelec, S.; Wichterle, H.; Woltjen, K.; Hughes, T. R.; Zandstra, P. W.; Nagy, A.; Wrana, J. L.; Blencowe, B. J. An Alternative Splicing Switch Regulates Embryonic Stem Cell Pluripotency and Reprogramming. *Cell* **2011**, *147* (1), 132–146.

(21) Medina, E.; Córdova, C.; Villalobos, P.; Reyes, J.; Komives, E. A.; Ramírez-Sarmiento, C. A.; Babul, J. Three-Dimensional Domain Swapping Changes the

Folding Mechanism of the Forkhead Domain of FoxP1. *Biophys. J.* **2016**, *110* (11), 2349–2360.

(22) Chu, Y. P.; Chang, C. H.; Shiu, J. H.; Chang, Y. T. Solution Structure and Backbone Dynamics of the DNA-binding Domain of FOXP1: Insight into Its Domain Swapping and DNA Binding. *Proteins* **2011**.

(23) Noel, J. K.; Whitford, P. C.; Onuchic, J. N. The Shadow Map: A General Contact Definition for Capturing the Dynamics of Biomolecular Folding and Function. *J. Phys. Chem. B* **2012**, *116* (29), 8692–8702.

(24) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11* (11), 5525–5542.

(25) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of Slow Molecular Order Parameters for Markov Model Construction. *J. Chem. Phys.* **2013**, *139* (1), 015102.

(26) Avila, C. L.; Drechsel, N. J. D.; Alcantara, R.; Villa-Freixa, J. Multiscale Molecular Dynamics of Protein Aggregation. *Current Protein & Peptide Science*. 2011, pp 221–234. https://doi.org/10.2174/138920311795860205.