OpenCafeMol: A GPU-accelerated coarse-grained biomolecular dynamics simulator with OpenMM library

Yutaka Murata[1], Toru Niina[2], Shoji Takada[1]

[1]Department of Biophysics, Graduate School of Science, Kyoto University, Kyoto 606-8502, Japan; [2]Laboratory for Computational Molecular Design, RIKEN Center for Biosystems Dynamics Research (BDR), 6-7-1 Minatojima-minamimachi, Chuo-ku, Kobe, Hyogo 650-0047 Japan

ABSTRACT

There has been an increasing demand for longer-timescale molecular dynamics (MD) simulations of larger biomolecular systems. To meet these demands, using the C++ API of OpenMM, we developed a fast and flexible MD software, OpenCafeMol, for residue-resolution protein and lipid models that shows high performance on graphics processing unit (GPU) machines. We validated OpenCafeMol for folding small proteins, lipid membrane dynamics, and membrane protein structures. Benchmark tests of the computation times showed that OpenCafeMol with one GPU for proteins and lipid membranes can be approximately 100 and 240 times faster than the corresponding simulations on a typical CPU machine (eight cores), respectively. We also illustrated the application of OpenCafeMol to vesicle fusion dynamics. The source code for OpenCafeMol is fully available.

INTRODUCTION

In recent years, there has been an increasing demand for longer-timescale molecular dynamics (MD) simulations of larger systems, especially in molecular biology. Rather than a single protein in aqueous solution, large protein complexes, crowded macromolecular systems, and biomolecular condensates have become the target of MD simulations. To meet these demands, two major approaches have been adopted: 1) speeding up computation using new computers and algorithms, and 2) coarse-graining the model representation. For the former, the special-purpose computer ANTON (1), general-purpose supercomputers (see, for example (2)), and graphics processing unit (GPU) (3, 4) have been successfully employed. For the latter, numerous coarse-grained models with different resolutions have been developed (5–16), along with methodologies to bridge different resolutions (17–21). There is no single universal coarse-grained model for biomolecules; however, different models with different resolutions are useful for different target systems.

Despite intensive studies on each of the two approaches, the combination of the two approaches has rarely been studied. Recently, GPU-accelerated coarse-grained (CG) MD simulations have been realized, showing promise over conventional CPU-based simulations. For this purpose, OpenMM provides a fast, flexible, and multiplatform framework to achieve such development (OpenMM 7 (22)). For example, OpenAWSEM and Open3SPN use the OpenMM Python library and show marked acceleration in CG protein and DNA simulations, respectively (23). OpenABC further extends this direction to extend its applicability (24).

In this study, along the same lines, we developed a fast and flexible MD simulator, called

OpenCafeMol, for residue-resolution coarse-grained models of proteins and lipids using the OpenMM library. OpenCafeMol implements a basic set of models used in the previously developed CafeMol software, such as AICG2+, iSoLF (25), and the HPS model (12, 26). For a few test systems of proteins, lipid membranes, and their mixtures, we performed MD simulations with OpenCafeMol, compared the results with those of CafeMol(27), and compared the computation time of OpenCafeMol with that of CPU-based software using typical laboratory-owned computers. The source code of the OpenCafeMol software is fully available on GitHub.

RESULTS

OpenCafeMol

OpenCafeMol is an OpenMM-based MD simulator used for CG biomolecular systems. Owing to its high performance and multiplatform OpenMM library, OpenCafeMol was developed for high-performance MD simulations, particularly for GPU-based simulations. Currently, it supports a few residue-resolution protein models and a lipid model that is compatible with the protein models.

Protein models that can be run with OpenCafeMol are Cα-based protein models, including an atomic interaction-based CG model AICG2+ (11), an original off-lattice Go-model developed by Clementi et al. (27), and an elastic-network model (28, 29). By preparing suitable input files, several additional CG models can be simulated using OpenCafeMol. For lipids, OpenCafeMol currently implements an original version of iSoLF(25) that represents one phospholipid molecule with five particles. In addition, several standard interactions were implemented, including the Debye–Hückel electrostatic interactions, the HPS model(26), (31)restraining potentials for the centers of mass of the two-particle groups, and the particle positions.

OpenCafeMol supports Langevin integrators with different ensembles such as NVT, NPT, and NPγT ensembles, in addition to simple energy minimization. Note that the GENESIS input assumes a time unit of picoseconds, whereas the TOML input assumes a time unit of one café-time = 49 fs (see the next paragraph). The recommended time steps were 0.01 (ps) with the GENESIS input, and 0.2 (café-time) with the TOML input.

OpenCafeMol was developed using the C++ API of the OpenMM Library layer (Fig. 1). It reads the input information using either TOML-style input files (defined in Mjolnir) or GENESIS input files (the latter is supported only partially at present). Potential energies were determined from the input information. Every potential energy term in CG models is encoded by the “class” of OpenCafeMol, which was defined by the use of the OpenMM class, as in Table I. Whenever necessary, users can add new types of potentials by creating a new class. Thus, it has a high expandability. To run the MD simulations, OpenCafeMol Simulator calls OpenMM Integrator and performs Langevin dynamics.

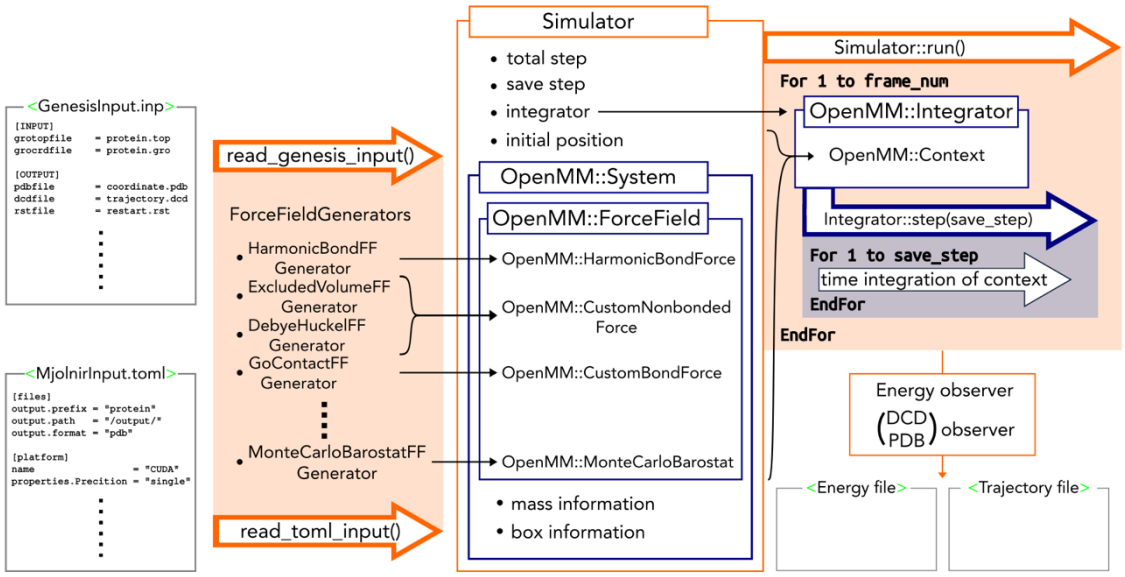

Figure 1 Schematic view of the OpenCafeMol software architecture.

Table I OpenCafeMol class*

	OpenCafeMol class**	OpenMM class
AICG2+ local 1st / iSoLF 1st	HarmonicBondFFG	HarmonicBondForce
AICG2+ local 2nd	GaussianBondFFG	CustomBondForce
AICG2+ local 3rd	GaussianDihedralFFG	CustomTorsionForce
AICG2+ local 4th	FlexibleLocalAngleFFG	CustomCompoundbondForce
AICG2+ local 5th	FlexibleLocalDihedralFFG	CustomTorsionForce
AICG2+ nonl 1st	GoContactFFG	CustomBondForce
AICG2+ nonl 2nd	ExcludedVolumeFFG	CustomNonbondedForce
Debye–Hückel	DebyeHuckelFFG	CustomNonbondedForce
iSoLF 2nd	HarmonicAngleFFG	HarmonicAngleForce
iSoLF 3rd / Protein-Lipid-Int / HPS	WeeksChandlerAndersenFFG	CustomNonbondedForce
iSoLF 4th	iSoLFAttractiveFFG	CustomNonbondedForce
Protein-Lipid-Int / HPS	LennardJonesAttractiveFFG	CustomNonbondedForce
	MonteCarloMembrane BarostatGenerator	MonteCarloMembraneBarostat
integrator	LangevinIntegratorGenerator	LangevinIntegrator

*) OpenCafeMol implements several more forcefield generators, which are included in the code.

**) FFG = ForceFieldGenerator in the code

From the user's perspective, OpenCafeMol has two interfaces: the GENESIS and TOML (Mjolnir) interfaces. The GENESIS interface is simple and its input format is essentially the same as that of GENESIS, which resembles the GROMACS input file format (30). Users who are unaware of the details of the force field can begin the simulation using this interface. The TOML-style input allows for more general types of potential that follow the file format of Mjolnir (https://github.com/Mjolnir-MD/Mjolnir). The TOML input file describes the detailed parameters of the force field so that users can fine-tune the force field. OpenCafeMol reads these two types of input files and translates them into the force field classes of OpenCafeMol, which generates OpenMM force fields.

Benchmark: protein folding with AICG2+ and electrostatic interaction

First, we evaluated OpenCafeMol for the folding of a small protein, the 56-residue Src homology 3 (SH3) domain, using the AICG2+ model of proteins with Debye-Hückel electrostatic interactions (11). The AICG2+ model is a structure-based residue-resolution CG protein model that has long been used in large-scale simulations. In addition, we added electrostatic interactions between charged residues (+1 for Arg and Lys, and -1 for Asp and Glu).

As expected, we observed temperature-dependent repeated folding and unfolding transitions, which were quantified as the time series of the Q-score, a fraction of the native contacts formed (Fig. 2a). The average Q-score curve as a function of temperature around the folding–unfolding transition is very sensitive to the model and simulation setup; thus, it can be used as a stringent test for MD simulators. We calculated the average Q-scores as a function of temperature using OpenCafeMol and CafeMol from 20 independent 10^7 MD step trajectories in each simulation (Fig. 2b). The denaturation curves from the two simulators agreed well, indicating that OpenCafeMol can accurately simulate protein folding.

We then measured and compared the calculation speed of OpenCafeMol with a GPU (RTX4090 1GPU) and the previously developed simulators, CafeMol.3.2.1, and GENESIS.1.7.1, with CPUs (Intel Xeon Gold 6226 1CPU, 8 cores). To examine the scalability of the simulations, we placed SH3 molecules in 3D cubic grids containing 1, 8, 27, 64, 125, and 216 SH3 molecules (the number of amino acids (particles) corresponded to 56, 448, 3584, 7000, and 12906, respectively). The computation time required for 10^6 time steps was measured (Fig. 2c). For one SH3 domain, OpenCafeMol was only 1.26 times faster than CafeMol, whereas for the largest tested case of 216 proteins, OpenCafeMol was ~100 times faster than CafeMol and GENESIS, which used CPUs.

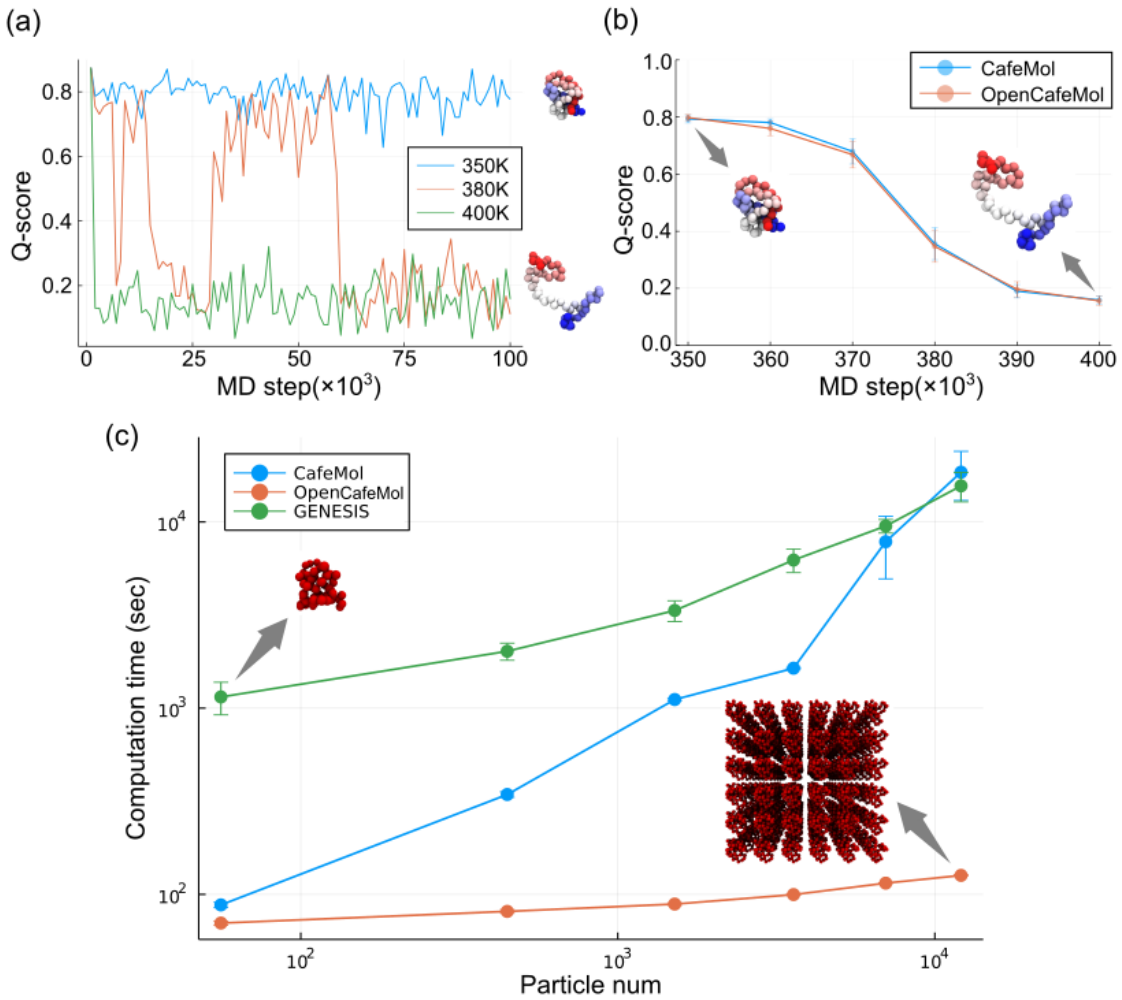

Figure 2 Benchmark test for folding of a small protein, the SH3 domain. (a) Representative time courses of Q-score at three temperatures. (b) The average Q-score as a function of temperature. Error bars represent the standard error. At each temperature, we ran 20 trajectories of 10^7 MD steps. The average Q-score was obtained from data at every 5.0×10^5 step in the latter half of each trajectory. (c) The benchmark of computation time for different size systems. Each data point contains 1 (56), 8 (448), 27 (1512), 64 (3584), 125 (7000), and 216 (12096) SH3 molecules (particles). The computation times were averaged over four trajectories. The error bar represents the standard error.

Benchmark for lipids: iSoLF model

Next, we tested OpenCafeMol for POPC lipid bilayer membrane dynamics using the CG lipid force field, iSoLF (25). The iSoLF model represents a POPC lipid with five particles: two particles for the head and three particles for the tails. In lipid membrane MD simulations, the area per lipid is sensitive to lipid dynamics, changing its mean value upon transition from the liquid-ordered phase to the liquid-disordered phase. Thus, the area per lipid molecule was used to test the simulation accuracy. We calculated the area per lipid of the POPC membrane in the temperature range of 270 K–400 K using OpenCafeMol and CafeMol (Fig. 3a). For each case, we performed 10^7 step MD simulations and sampled structures every 10^5 steps after the initial equilibration of 5.0×10^6 steps. The area per lipid as a function of temperature agreed well between OpenCafeMol and CafeMol. We conclude that the iSoLF model in OpenCafeMol is accurate.

We then benchmarked the computation time of the iSoLF model using OpenCafeMol with GPUs and CafeMol with CPUs (Fig. 3b). For this purpose, we used a series of square POPC lipid bilayer membranes containing 128, 200, 800, 1800, 3200, and 5000 POPC molecules. The numbers of particles in each case were 640, 1000, 4000, 9000, 16000, and 20000. Using the same hardware as above, the calculation speed of the GPU-based OpenCafeMol was 14.1 times faster than that of the CPU-based CafeMol for the system with 128 lipids, whereas the speed-up ratio reached 239 times that of the 5,000-lipid system.

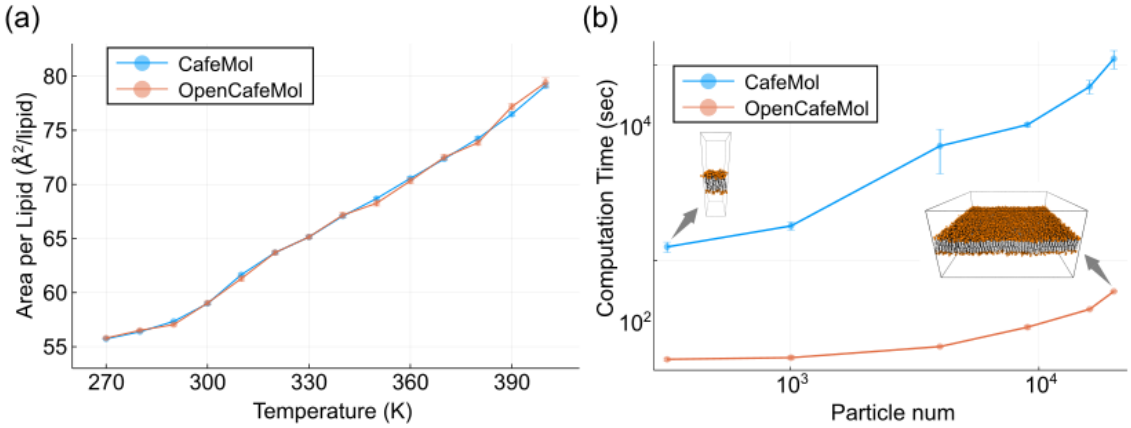

Figure 3 Benchmark test for the POPC lipid bilayer membrane. (a) Area per lipid as a function of temperature. For each point, the data point is the average of every 10^5 steps after the initial 5.0×10^7 step from 10^8 MD step simulations. Error bars correspond to standard errors. (b) The benchmark result of POPC membrane simulation.

Benchmark for protein in a lipid membrane

Third, we examined OpenCafeMol as a membrane protein, rhodopsin (PDBID:5AX0), embedded in a POPC lipid membrane using the iSoLF model and the protein-lipid interaction developed by Ugarte et al. (31). The test system comprised a unit that included one rhodopsin molecule embedded in 94 POPC molecules (Fig. 4a, left). First, we examined the distribution of rhodopsin tilt angle with respect to the membrane plane (Fig. 4ab). The overall distribution of OpenCafeMol was consistent with that of CafeMol. The distribution was slightly sharper, which can be attributed to a slight difference in the surface tension of the membrane.

To benchmark the computation time, we prepared a series of systems containing 1, 4, 9, 16, and 25 units. In the case with one unit, OpenCafeMol was just 3.6 times faster than CafeMol, whereas the speed-up ratio reached 63 for the 25-unit system.

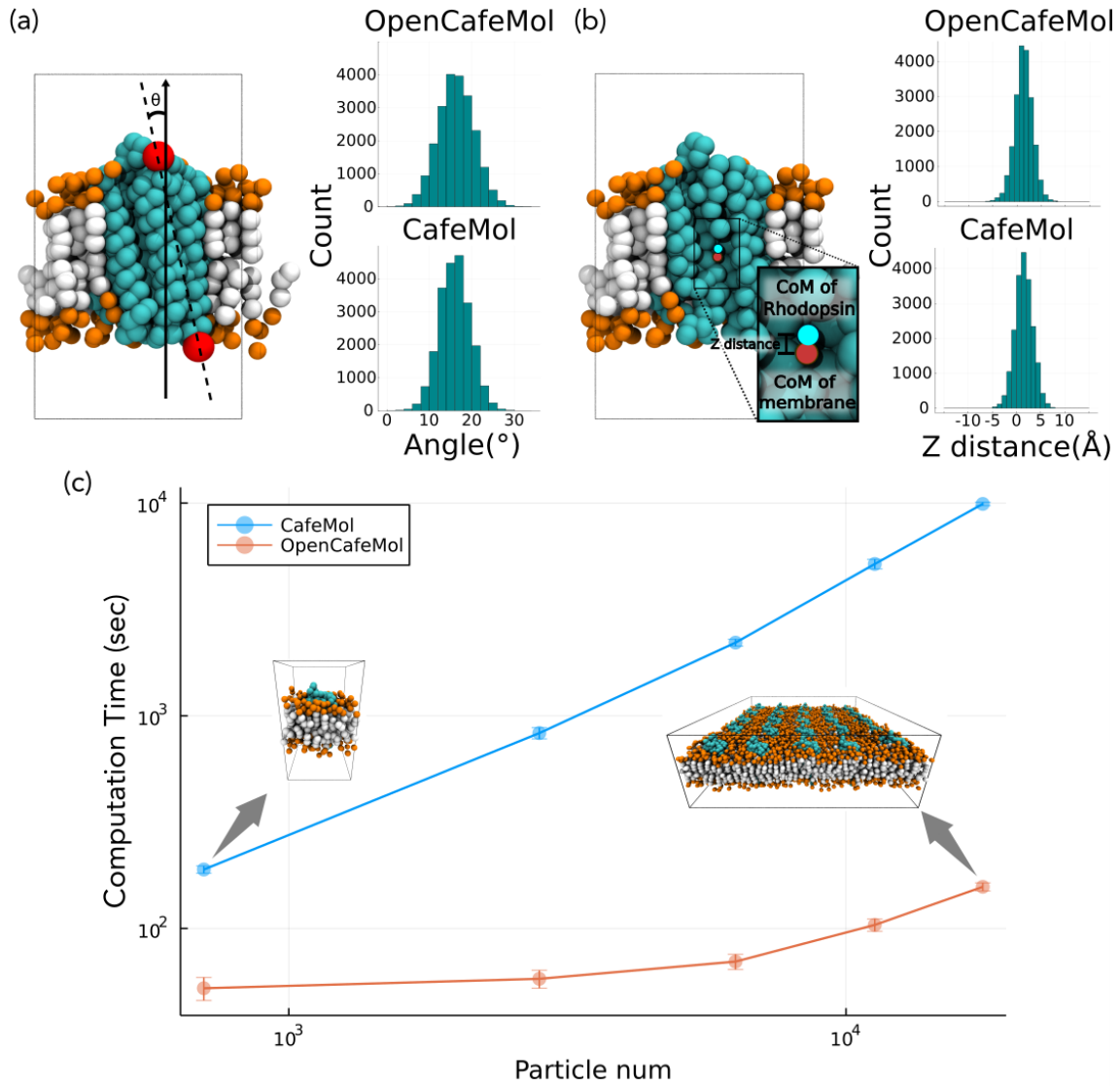

Figure 4 Benchmark text for a protein-lipid mixture. (a) Distribution of rhodopsin tilt angle to the normal of the membrane surface. Top: OpenCafeMol; bottom: CafeMol. (b) Distribution of rhodopsin insertion to the membrane. Top: OpenCafeMol; bottom: CafeMol. (c) Computation time of lipid-protein system simulation. Each data point corresponds to 1 (704), 4 (2816), 9 (6336), 16 (11264), and 25 (17600) units (particles).

Vesicle merge simulation

Finally, we used OpenCafeMol to simulate membrane fusion of the two liposomes as an additional demonstration. Each liposome is composed of 5710 POPC molecules, the diameter of which is about 50 Å (Fig. 5a). To induce membrane fusion, we set pseudo-particles at the center of each liposome and pulled them using a harmonic potential for inter-pseudo-particle distance, with no short-range repulsion between the two pseudo-particles. We illustrate a representative trajectory of the fusion process. Two vesicles made contact around the 8.0×10^5-th MD step, which was followed by the fusion of two vesicles around the 1.6×10^6-th MD step (Fig. 5bc). The peanut-shaped fused vesicle slowly changed shape to a near-spherical form during the subsequent 10^7 MD steps. The homogenization of lipids took a rather long time and was completed near the end

of 10^8 MD step simulations (Fig. 5bc).

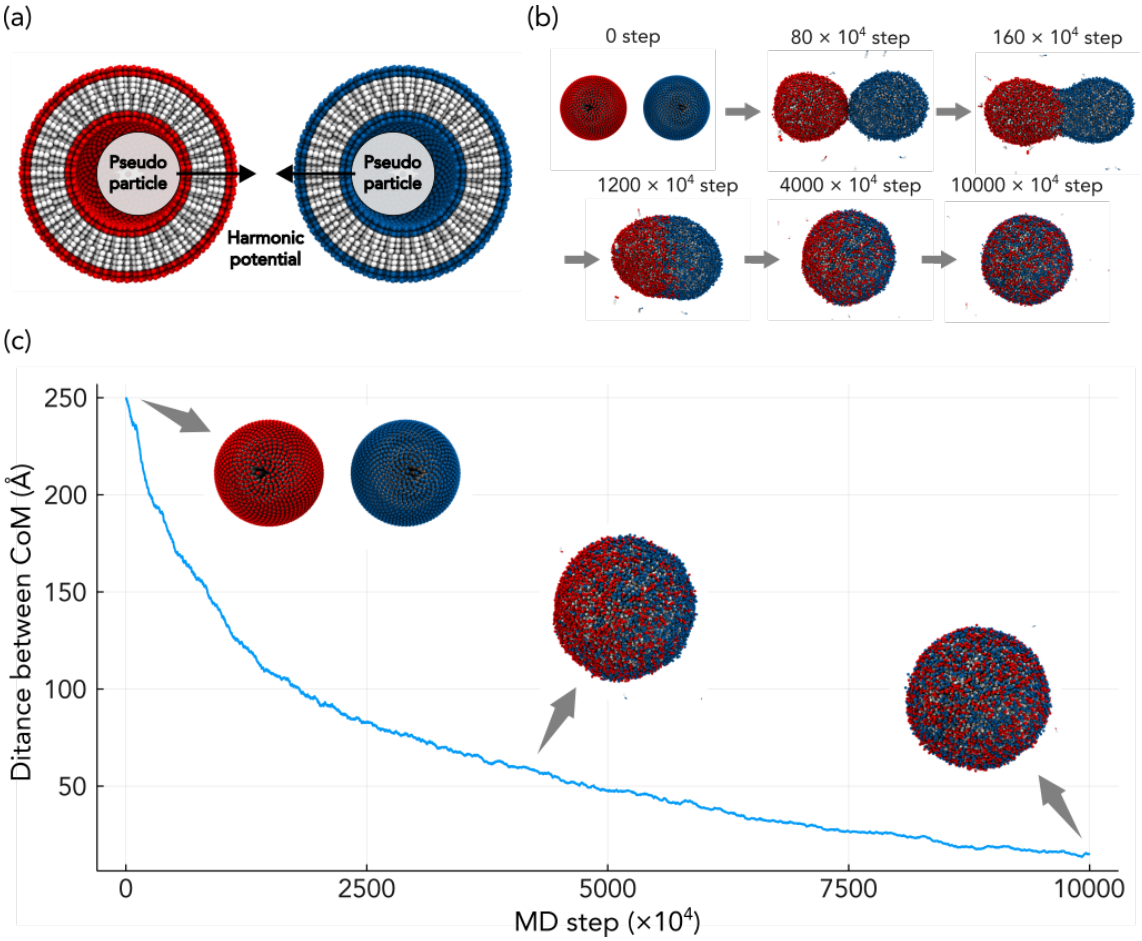

Figure 5 Vesicle fusion simulation. (a) The system is composed of two vesicles each with one pseudo-particle in the central cavity, used for pulling. (b) Snapshots of vesicle fusion. (c) A representative trajectory of the liposome merging process. The vertical axis is the distance between the centers of masses of lipids initially contained in two (red and blue) vesicles.

METHODS

The protein model AICG2+

The force field used for the protein benchmark was the AICG2+ model and the Debye–Hückel electrostatic interaction(11). The AICG2+ is a structure-based CG model where one amino acid is represented by one CG particle located on its Cα atom. This can be expressed using the following function:

$$V_{\mathrm{AICG2+}} = V_{\mathrm{local}} + V_{\mathrm{nonlocal}}$$

where V_{local} represents the potential related to neighboring particles connected by covalent bonds and can be written as follows:

$$V_{\text{local}}(\Gamma) = \sum_{b_i \in \text{bonds}} k_{b,i}\left(b_i - b_{i,0}\right)^2 - \sum_{\theta_i \in \text{angles}} k_B T \ln \frac{P_\theta(\theta_i|i)}{\sin\theta_i} - \sum_{\varphi_i \in \text{dihedrals}} k_B T \ln P_d(\varphi_i|i)$$
$$- \sum_{r_i \in 1-3\text{pairs}} \varepsilon_i \exp\left(\frac{-\left(r_i - r_{i,0}\right)^2}{2w_i^2}\right) - \sum_{\varphi_i \in \text{dihedrals}} \varepsilon_{\varphi,i} \exp\left(\frac{-\left(\varphi_i - \varphi_{i,0}\right)^2}{2\sigma_{\varphi,i}^2}\right)$$

Here, Γ collectively represents the coordinates of all the particles in the target protein. The first term represents the harmonic potential required to limit the virtual bond lengths. The second and third terms are statistical potentials for virtual bond angles and virtual dihedral angles, respectively, representing amino acid sequence-dependent properties, where $P_\theta(i)$ and $P_d(i)$ are probability distributions of angles and dihedral angles that depend on local amino acid sequences. The fourth and fifth terms represent native structure-specific local interactions for the so-called 1-3 pairs and dihedral angles, respectively. V_{nonlocal} is the potential for nonlocal particles along the chain and is represented as

$$E_{\text{nonlocal}}(r_{ij}) = \varepsilon_{\text{Go},i,j}\left[5\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - 6\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{10}\right] + E_{\text{exv}}(r_{ij})$$

The first term is structure-based Gō-like potential, where r_{ij} is the distance between residues that are interacting with each other in the native structure. σ_{ij} is the corresponding distance in the native structure. $\varepsilon_{\text{Go},i,j}$ is the energy coefficient that depends on atomic interaction between two residues at the native structure. The second term is an excluded volume potential and can be written as follows:

$$E_{\text{exv}}(r_{ij}) = \begin{cases} -\varepsilon_{\text{exv}}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} + \epsilon'_{\text{exv},ij}, r_{ij} < r_C \\ 0, r_{ij \geq} r_c \end{cases}$$

σ_{ij} is excluded volume distance and depends on residue type. ε_{exv} is force coefficient and 0.6 kcal/mol, r_C is the cutoff distance and corresponds to $2\sigma_{ij}$, $\epsilon'_{\text{exv},ij}$ is a correction for the cutoff and corresponds to $(1/2)^{12}\varepsilon_{\text{exv}}$.

The Debye–Hückel electrostatic interaction can be written as following function.

$$E_{\text{debye}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_k r_{ij}} \mathrm{e}^{-r_{ij}/\lambda_D}$$

r_{ij} is the distance between two non-bonded charged particles and ϵ_0 is the dielectric permittivity of vacuum. ϵ_k is the relative permittivity of the solution. λ_D is the Debye length and is defined by $\lambda_D = \sqrt{k_B T \epsilon_0 \epsilon_k / (2N_A e_c^2 I)}$, where N_A is the Avogadro's number, e_c is the elementary charge, and I is the ionic strength of the solution.

For the protein simulation, we used the Langevin integrator in OpenMM and 0.01 cafetime^{-1} for the friction coefficient of all particles. The default integrator and friction coefficients were used in GENESIS and CafeMol. We used 0.1 cafetime as the MD step size Δt with the TOML input interface.

The membrane model iSoLF

For the simulation including the membrane, we used the iSoLF force field for lipids(25). The iSoLF force field can be expressed as follows:

$$V_{\mathrm{iSoLF}}(\Gamma) = \sum_{b_i \in \mathrm{bonds}} k_{b,i}\left(b_i - b_{0,i}\right)^2 + \sum_{\theta_i \in \mathrm{angles}} k_{\theta,i}\left(\theta_i - \theta_{0,i}\right)^2 + V_{\mathrm{repulsion}}(\Gamma) + V_{\mathrm{attraction}}(\Gamma)$$

The first and second terms are the harmonic potentials to restrain the virtual bond lengths and the harmonic potential for the virtual bond angles, respectively. $V_{\mathrm{repulsion}}$ is the Weeks–Chandler Andersen (WCA) potential as a short-range repulsion; explicitly, it has the formula:

$$V_{\mathrm{repulsion}}(\Gamma) = \sum_{(i,j) \in \mathrm{non-bonded\ pairs}} \begin{cases} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6} + \frac{1}{4}\right], r_{ij} \leq \sqrt[6]{2}\sigma_{ij} \\ 0, r_{ij} > \sqrt[6]{2}\sigma_{ij} \end{cases}$$

where σ_{ij} is the arithmetic mean $\left(\sigma_i + \sigma_j\right) / 2$ and ε_{ij} is the geometric mean $\sqrt{\varepsilon_i \varepsilon_j}$. $V_{\mathrm{attraction}}$ is a short-range attraction potential and can be expressed as follows:

$$V_{\mathrm{attraction}}(\Gamma) = \sum_{(i,j) \in \mathrm{non-bonded\ pairs}} \begin{cases} -\varepsilon_{ij}, r_{ij} \leq \sqrt[6]{2}\sigma_{ij} \\ -\varepsilon_{ij}\cos^2\left[\frac{\pi}{2\omega_{ij}}\left(r_{ij} - \sqrt[6]{2}\sigma_{ij}\right)\right], \sqrt[6]{2}\sigma_{ij} < r_{ij} \leq \sqrt[6]{2}\sigma_{ij} + \omega_{ij} \\ 0, \sqrt[6]{2}\sigma_{ij} + \omega_{ij} < r_{ij} \end{cases}$$

where ω_{ij} is the range over which the particles are subject to attractive force.

The integrator in the CafeMol simulation for the iSoLF is the method developed by Gao et al. (32). The barostat was semi-isotropic pressure coupling for XY direction and Z was fixed, called $N\gamma_{xy}L_zT$ ensemble. For OpenCafeMol, we used the same integrator as that of the protein, and the friction coefficient was 0.005 cafetime^{-1} for all particles. We used the periodic boundary box and MonteCarloMembraneBarostat with XYIsotropic, Z fixed. For the lipid-only simulation, Δt is 0.1 cafetime. In initial configuration, all the lipids are arranged in a square pattern with equal spacing and the box size was tuned so that the APL is 64Å^2/lipid.

To analyze the tilt angle of rhodopsin in the membrane, we defined the tilt angle as the angle between the z-axis and the line between the rhodopsin N-terminal- and C-terminal particles.

Protein-lipid interactions

For the protein and lipid simulations, we used the lipid-protein interaction force field developed by Ugarte et al. (31). The force field is expressed as follows:

$$V_{\mathrm{LPI}} = V_{rep} + V_{HP}$$

V_{rep} is the repulsive term and modeled by WCA potential in the same way as iSoLF forcefield, except for $\sigma_{ij} = \sqrt[6]{(\sigma_i^6 + \sigma_j^6)/2}$. V_{HP} is the hydrophobic-hydrophilic potential and can be expressed as follows:

$$V_{HP}(\Gamma) = \sum_{(i,j) \in LPI-pairs} \begin{cases} -\varepsilon_{HP,ij}, r_{ij} \leq \sqrt[6]{2}\sigma_{ij} \\ 4\varepsilon_{HP,ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right], r_{ij} > \sqrt[6]{2}\sigma_{ij} \end{cases}$$

σ_{ij} is the same as of V_{rep}. $\varepsilon_{HP,ij}$ is specific value for the particle pairs. In this model, each lipid

was composed of five particles and 20 types of natural amino acids for a total of 100 parameters. We used 0.2 cafetime as Δt. The initial conditions were prepared using the following steps. First, 128 lipid molecules were prepared in the same manner as for the membrane benchmark. Rhodopsin was then placed on the membrane, and the overlapping lipids were removed. This system was treated as one unit and aligned with 4, 9, 16, and 25 units in the shape of a square. Before the production run, the initial configuration was equilibrated with 1000 steps under NVT conditions.

HPS model

The HPS model(26) was implemented for simulation of interacting disordered regions in OpenCafeMol. The potential function for this model is expressed as follows:

$$V_{HPS} = V_{rep} + V_{attr}$$

V_{rep} is the repulsive term and modeled by WCA potential in the same way as iSoLF forcefield, and V_{attr} is hydrophobic-hydrophilic potential and can be modeled by the same formulation of V_{HP} in the protein-lipid interaction.

Vesicle merge simulation

For the vesicle simulation, we first built a vesicle configuration by aligning lipids on the sphere, of which the inner radius is 50 Å with 64 Å^2 area per lipid. Then, we placed two vesicle 250 Å apart. We put a pseudo sphere at the center of each vesicle with the WCA repulsive force to lipid particles; the parameter of WCA potential is $\sigma_{\mathrm{pseudo}} = 40$ Å, $\varepsilon_{\mathrm{pseudo}} = 10.0$ kcal/mol. To pull one vesicle toward the other, we applied a spring force as follows:

$$V = \frac{1}{2}k\left(r_{IJ} - r_0\right)^2$$

where I and J are centers of mass of particles in groups I and J, and $k = 0.01$ kcal/mol is the spring constant for pulling.

DISCUSSION AND CONCLUSIONS

In this study, we developed a fast and flexible MD software, OpenCafeMol, for residue-resolution protein and lipid models, which exhibited high performance on GPU machines. We validated OpenCafeMol for the folding of a small protein using the AICG2+ model, lipid membrane dynamics using the iSoLF model, and membrane protein structure. Benchmark tests of the computation times showed that OpenCafeMol with one GPU (RTX4090) for proteins and lipid membranes was approximately 100 and 240 times faster than the corresponding simulations with a typical CPU (Intel Xeon Gold 6226, 1CPU, 8 cores), respectively.

As new software, there is much room for extension in OpenCafeMol. A DNA model that is compatible with the residue-resolution protein model is highly desirable. In a separate article, we are reporting the implementation of the 3SPN.2C model of DNA; the 3SPN.2C model approximates one nucleotide as three particles, each representing phosphate, sugar, and nucleobase, and has broadly been used in the community (33). In addition, we can simulate protein-DNA complexes with OpenCafeMol on a GPU, which is orders of magnitude faster than CafeMol on a CPU. For proteins, we implemented a structure-based model, such as AICG2+, for protein dynamics simulations in one state. Often, to simulate transitions between two or more

states, multiple-basin models are useful (34) but have not been implemented. For lipid models, we implemented only the first version of iSoLF; however, a much more extended version has been developed (35) that needs to be implemented. For time integration and structural sampling, we implemented only the Langevin dynamics at a constant temperature. Therefore, the implementation of advanced sampling methods is highly desirable. Currently, the GENESIS input files are only partially supported. We are working to support the GENESIS input files.

AUTHOR CONTRIBUTIONS

Y.M. and S.T. conceived the project, and Y.M. and T.N. developed the code. Y.M. performed the simulations, analyzed the data, and assembled the figures; Y.M. and S.T. discussed the results and wrote the manuscript.

ACKNOWLEDGMENTS

We would like to thank Masataka Yamauchi for helpful discussions. This work was supported by the JSPS KAKENHI grants 20H05934 (S.T.), 21H02441 (S.T.), and 24K01991 (S.H.) by the MEXT grant JPMXP1020230119 as "Program for Promoting Researches on the Supercomputer Fugaku" (S.T.)

REFERENCES

1. Shaw, D.E., M.M. Deneroff, R.O. Dror, J.S. Kuskin, R.H. Larson, J.K. Salmon, C. Young, B. Batson, K.J. Bowers, J.C. Chao, M.P. Eastwood, J. Gagliardo, J.P. Grossman, C.R. Ho, D.J. Ierardi, I. Kolossváry, J.L. Klepeis, T. Layman, C. McLeavey, M.A. Moraes, R. Mueller, E.C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, and S.C. Wang. 2007. Anton, a special-purpose machine for molecular dynamics simulation. In: Proceedings of the 34th annual international symposium on Computer architecture. New York, NY, USA: ACM. pp. 1–12.
2. Yoshida, T. 2018. Fujitsu High Performance CPU for the Post-K Computer. *IEEE Hot Chips 30 Symposium*. 21.
3. Páll, S., A. Zhmurov, P. Bauer, M. Abraham, M. Lundborg, A. Gray, B. Hess, and E. Lindahl. 2020. Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. *J Chem Phys*. 153:134110.
4. Glaser, J., T.D. Nguyen, J.A. Anderson, P. Lui, F. Spiga, J.A. Millan, D.C. Morse, and S.C. Glotzer. 2015. Strong scaling of general-purpose molecular dynamics simulations on GPUs. *Comput Phys Commun*. 192:97–107.
5. Takada, S., R. Kanada, C. Tan, T. Terakawa, W. Li, and H. Kenzaki. 2015. Modeling Structural Dynamics of Biomolecular Complexes by Coarse-Grained Molecular Simulations. *Acc Chem Res*. 48:3026–3035.
6. Kmiecik, S., D. Gront, M. Kolinski, L. Wieteska, A.E. Dawid, and A. Kolinski. 2016. Coarse-Grained Protein Models and Their Applications. *Chem Rev*. 116:7898–7936.
7. Souza, P.C.T., R. Alessandri, J. Barnoud, S. Thallmair, I. Faustino, F. Grünewald, I. Patmanidis, H. Abdizadeh, B.M.H. Bruininks, T.A. Wassenaar, P.C. Kroon, J. Melcr, V.

Nieto, V. Corradi, H.M. Khan, J. Domański, M. Javanainen, H. Martinez-Seara, N. Reuter, R.B. Best, I. Vattulainen, L. Monticelli, X. Periole, D.P. Tieleman, A.H. de Vries, and S.J. Marrink. 2021. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nat Methods*. 18:382–388.

8. Darré, L., M.R. Machado, A.F. Brandner, H.C. González, S. Ferreira, and S. Pantano. 2015. SIRAH: A structurally unbiased coarse-grained force field for proteins with aqueous solvation and long-range electrostatics. *J Chem Theory Comput*. 11:723–739.
9. Klein, F., M. Soñora, L. Helene Santos, E. Nazareno Frigini, A. Ballesteros-Casallas, M. Rodrigo Machado, and S. Pantano. 2023. The SIRAH force field: A suite for simulations of complex biological systems at the coarse-grained and multiscale levels. *J Struct Biol*. 215:107985.
10. Davtyan, A., N.P. Schafer, W. Zheng, C. Clementi, P.G. Wolynes, and G.A. Papoian. 2012. AWSEM-MD: Protein structure prediction using coarse-grained physical potentials and bioinformatically based local structure biasing. *Journal of Physical Chemistry B*. 116:8494–8503.
11. Li, W., W. Wang, and S. Takada. 2014. Energy landscape views for interplays among folding, binding, and allostery of calmodulin domains. *Proc Natl Acad Sci U S A*. 111:10550–10555.
12. Dignon, G.L., W. Zheng, Y.C. Kim, R.B. Best, and J. Mittal. 2018. Sequence determinants of protein phase behavior from a coarse-grained model. *PLoS Comput Biol*. 14:e1005941.
13. Her, C., T.M. Phan, N. Jovic, U. Kapoor, B.E. Ackermann, A. Rizuan, Y.C. Kim, J. Mittal, and G.T. Debelouchina. 2022. Molecular interactions underlying the phase separation of HP1α: role of phosphorylation, ligand and nucleic acid binding. *Nucleic Acids Res*. 50:12702–12722.
14. Cao, F., S. von Bülow, G. Tesei, and K. Lindorff-Larsen. 2024. A coarse-grained model for disordered and multi-domain proteins. *Protein Science*. 33:2024--02.
15. Freeman, G.S., D.M. Hinckley, J.P. Lequieu, J.K. Whitmer, and J.J. de Pablo. 2014. Coarse-grained modeling of DNA curvature. *J Chem Phys*. 141:165103.
16. Jung, J., C. Tan, and Y. Sugita. 2024. GENESIS CGDYN: large-scale coarse-grained MD simulation with dynamic load balancing for heterogeneous biomolecular systems. *Nat Commun*. 15:3370.
17. Izvekov, S., and G.A. Voth. 2005. A Multiscale Coarse-Graining Method for Biomolecular Systems. *J Phys Chem B*. 109:2469–2473.
18. Izvekov, S., and G.A. Voth. 2005. Multiscale coarse graining of liquid-state systems. *J Chem Phys*. 123:134105.
19. Noid, W.G., J.-W. Chu, G.S. Ayton, and G.A. Voth. 2007. Multiscale Coarse-Graining and Structural Correlations: Connections to Liquid-State Theory. *J Phys Chem B*. 111:4116–4127.
20. Noid, W.G., P. Liu, Y. Wang, J.-W. Chu, G.S. Ayton, S. Izvekov, H.C. Andersen, and G.A. Voth. 2008. The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *J Chem Phys*. 128:244115.
21. Noid, W.G., J.-W. Chu, G.S. Ayton, V. Krishna, S. Izvekov, G.A. Voth, A. Das, and H.C. Andersen. 2008. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J Chem Phys*. 128:244114.
22. Eastman, P., J. Swails, J.D. Chodera, R.T. McGibbon, Y. Zhao, K.A. Beauchamp, L.P. Wang, A.C. Simmonett, M.P. Harrigan, C.D. Stern, R.P. Wiewiora, B.R. Brooks, and V.S. Pande. 2017. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput Biol*. 13:1–17.
23. Lu, W., C. Bueno, N.P. Schafer, J. Moller, S. Jin, X. Chen, M. Chen, X. Gu, A. Davtyan, J.J. de Pablo, and P.G. Wolynes. 2021. OpenAWSEM with Open3SPN2: A fast, flexible, and accessible framework for large-scale coarse-grained biomolecular simulations. *PLoS*

Comput Biol. 17:e1008308.
24. Liu, S., C. Wang, A.P. Latham, X. Ding, and B. Zhang. 2023. OpenABC enables flexible, simplified, and efficient GPU accelerated simulations of biomolecular condensates. *PLoS Comput Biol*. 19:e1011442.
25. La Torre, D.U., and S. Takada. 2020. Coarse-grained implicit solvent lipid force field with a compatible resolution to the C α protein representation. *Journal of Chemical Physics*. 153:205101.
26. Tesei, G., T.K. Schulze, R. Crehuet, and K. Lindorff-Larsen. 2021. Accurate model of liquid-liquid phase behavior of intrinsically disordered proteins from optimization of single-chain properties. *Proc Natl Acad Sci U S A*. 118:e2111696118.
27. Clementi, C., H. Nymeyer, and J.N. Onuchic. 2000. 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*. 298:937–953.
28. Bahar, I., A.R. Atilgan, and B. Erman. 1997. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold Des*. 2:173–181.
29. Hinsen, K. 1998. Analysis of domain motions by approximate normal mode calculations. *Proteins: Structure, Function, and Genetics*. 33:417–429.
30. Abraham, M.J., T. Murtola, R. Schulz, S. Páll, J.C. Smith, B. Hess, and E. Lindah. 2015. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*. 1–2:19–25.
31. Ugarte La Torre, D., and S. Takada. 2021. Modeling lipid-protein interactions for coarse-grained lipid and Cα protein models. *Journal of Chemical Physics*. 155:155101.
32. Gao, X., J. Fang, and H. Wang. 2016. Sampling the isothermal-isobaric ensemble by Langevin dynamics. *Journal of Chemical Physics*. 144:124113.
33. Freeman, G.S., D.M. Hinckley, and J.J. de Pablo. 2011. A coarse-grain three-site-per-nucleotide model for DNA with explicit ions. *J Chem Phys*. 135:165104.
34. Okazaki, K.I., N. Koga, S. Takada, J.N. Onuchic, and P.G. Wolynes. 2006. Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations. *Proc Natl Acad Sci U S A*. 103:11844–11849.
35. Ugarte La Torre, D., S. Takada, and Y. Sugita. 2023. Extension of the iSoLF implicit-solvent coarse-grained model for multicomponent lipid bilayers. *Journal of Chemical Physics*. 159:075101.