Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Structural Biology

Adaptive ensemble simulations of biomolecules

Peter M Kasson[1,2] and Shantenu Jha[3,4]

Recent advances in both theory and computational power have created opportunities to simulate biomolecular processes more efficiently using adaptive ensemble simulations. Ensemble simulations are now widely used to compute a number of individual simulation trajectories and analyze statistics across them. Adaptive ensemble simulations offer a further level of sophistication and flexibility by enabling high-level algorithms to control simulations-based on intermediate results. We review some of the adaptive ensemble algorithms and software infrastructure currently in use and outline where the complexities of implementing adaptive simulation have limited algorithmic innovation to date. We describe an adaptive ensemble API to overcome some of these barriers and more flexibly and simply express adaptive simulation algorithms to help realize the power of this type of simulation.

Addresses
[1] Departments of Molecular Physiology and of Biomedical Engineering, University of Virginia, Charlottesville, VA 22908, United States
[2] Science for Life Laboratory, Department of Cell and Molecular Biology, Uppsala University, Uppsala 75146, Sweden
[3] Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, United States
[4] Center for Data-Driven Discovery, Brookhaven National Laboratory, Upton, NY 11793, United States

Corresponding authors: Kasson, Peter M (kasson@virginia.edu), Jha, Shantenu (shantenu.jha@rutgers.edu)

Current Opinion in Structural Biology 2018, **52**:87–94

This review comes from a themed issue on **Biophysical and computational methods**

Edited by **Gregory Voth** and **Mark Yeager**

https://doi.org/10.1016/j.sbi.2018.09.005

0959-440X/© 2018 Elsevier Ltd. All rights reserved.

Introduction

In recent years, molecular dynamics simulation has been increasingly utilized for quantitative prediction of and insight into biophysical problems. Moving beyond visualization and qualitative insight, robust statistical estimation of thermodynamic, kinetic, or structural properties of biomolecules is now within reach. However, this new horizon creates a set of challenges, since statistical estimation of these quantities requires many simulated 'observations' of the desired process and quantitative insight thus comes at a cost of substantial computational requirements. Much like single-molecule experiments, the most efficient way to obtain these simulated observations is using collections or *ensembles* of simulations rather than a single extremely long trajectory of single-event observations [1,2]. Such ensemble approaches have also been used to predict effects of mutations at different sites on a protein, to estimate transition states and free-energy barriers, and make other similar quantitative predictions [3–6,7•,8•,9,10].

A further advance in the range of biomolecular processes that can be feasibly simulated comes from not only simulating biomolecular ensembles but running these simulations in an *adaptive* manner, where high-level algorithms are used to determine the next round of simulations-based on the results of the previous one. Such adaptive algorithms can increase simulation efficiency by greater than a thousand-fold [11,12••,13] but require a more sophisticated software infrastructure to support them. Here, we review some of the biophysical insights gained via ensemble simulations, the software needs and current capabilities for flexibly and efficiently running such calculations, and a pathway to filling some of the unmet needs in this area.

Adaptive ensemble methods for simulating biomolecules

Although frameworks for adaptive ensemble simulation have been developed only recently [12••,14••,15•], simpler algorithms for adaptive ensemble simulation have been in use for many years. Many algorithms pre-specify the sequence of computational simulations, but the results of each set of simulations are used to determine the inputs for the next round. Algorithms where not only the simulation parameters but even the type of computational operation to perform depends on intermediate results are rarer, due in large part to the higher barrier to implementing such algorithms, but these present perhaps the most exciting and powerful set of simulation approaches.

Replica exchange molecular dynamics is a long-standing and widely used ensemble method where individual simulations within an ensemble exchange coordinates over the course of the simulation. Depending on the exchange algorithm, this can be performed in either a non-adaptive or an adaptive fashion. Replica exchange was originally formulated as temperature replica exchange [16–18], where an ensemble of simulations is run at different temperatures to facilitate escape from energy minima, exchanging coordinates via a Monte

Carlo criterion. In a related method, generalized-ensemble simulation, exchange can be performed over larger numbers of generalized 'coordinates', including between different Hamiltonians, and different exchange algorithms between ensemble members can be employed [19–21]. This has permitted exploration of free-energy surfaces that are less accessible to temperature replica exchange alone. One example of increased adaptivity in such simulations comes from adaptive placement of scaling parameters ('lambda values') in free-energy perturbation and similar calculations to optimize statistical convergence [22–26,27•]. Expanded-ensemble simulations are related to replica exchange; in terms of parallelization, they can be seen as serial adaptive algorithms that can be parallelized whereas replica exchange is an ensemble algorithm that can be made adaptive. Adaptivity in these cases has largely been supported via explicit implementation in molecular dynamics software packages, and a more flexible platform for such adaptive algorithms would potentiate further algorithmic development, reuse of existing algorithms by other scientists, and resulting scientific progress. Conversely, metadynamics approaches have been implemented primarily in high-level software such as PLUMED [28,29•] that abstracts the adaptivity for algorithms such as multiple-walker metadynamics [30] but requires explicit job scheduling.

Adaptive ensemble simulation has been particularly helpful in biomolecular simulation algorithms where each individual simulation uses an identical unbiased Hamiltonian but where placement of simulations in phase space is optimized to improve estimation of the kinetics and thermodynamics of a biomolecular process. In an ensemble formulation, placement of unbiased trajectories in phase space involving choosing which trajectories to extend or from which already-sampled points in position and velocity space to start new trajectories. Some of these unbiased-trajectory algorithms include milestoning, weighted-ensemble simulation, and related techniques [15•,31,32,33•,34]; each of these has been implemented in custom software packages to facilitate the adaptive logic and post-processing.

Methods to construct Markov State Models from molecular dynamics simulations provide a similar set of powerful approaches for analyzing molecular kinetics using unbiased individual trajectories [35–38]. The choice of starting points for these trajectories can be optimized to reduce the uncertainty of the resulting model: it has been demonstrated retrospectively and then prospectively that adaptive sampling with Markov State Models increases convergence efficiency by several orders of magnitude. Adaptive sampling methods have recently been applied with great success to complex biomolecular processes [39••]. Another recent study combines biased umbrella sampling simulations with Markov State Model-inspired estimators and adaptive sampling, showing how the facile combination of methods can potentiate further insight [40•]. However, this can be difficult because most implementations of such methods have been in special-purpose code. One exception is Copernicus [12••,27•], but that has other limitations as detailed below.

Designing software systems for adaptive ensemble methods

The broad range of adaptive ensemble simulation algorithms impose similarly diverse requirements on the underlying software infrastructure. Algorithms differ in the frequency of communication between ensemble members, local versus non-local communication, and the type of information exchanged. Two adaptive simulation work/data flow diagrams are schematized in Figure 1. Adaptive changes can alter the number of tasks being performed (how many ensemble members in a simulation), the parameters of those tasks (placement of temperature or lambda values in an expanded-ensemble simulation), or even which tasks are being performed when (e.g. branching between simulations to converge a bound-complex ensemble and free-energy-perturbation simulations to measure binding of a new candidate ligand and either accept or reject that ligand for inclusion into the main simulation loop). The logic to specify such changes can rely on a single simulation within an ensemble, an operation across an ensemble, or even external criteria, such as changes in resource availability or new experimental data.

Despite this diversity, a key commonality among adaptive algorithms is that they can be expressed at a high level, such that the adaptive logic itself is independent of simulation details. This separation of adaptive operations from simulation internals provides a useful and important abstraction for both methods developers and the software system. Adaptive operations that are expressed independent of the internal details of tasks facilitate MD software package agnosticism and simpler expression of different types of adaptivity and responses to adaptivity. This promotes facile development of new methods while facilitating optimization and performance engineering that will be needed at large scales.

Expressing adaptive algorithms in this more abstract manner, as computational processes separate from but operating on independent ensemble members, creates several implementation challenges. These include coordination and consistency across distributed execution components, scalable communication between independent simulations and efficient stop and restart of simulations. Separating the adaptive logic from underlying execution management software allows the complexity to be contained within the internal implementation of the software system and not be exposed to the user. This approach also enables transparent low-level optimization