Overcoming Sampling Issues and Improving Computational Efficiency in Collective-Variable-Based Enhanced-Sampling Simulations: A Tutorial

Haohao Fu, Mengchen Zhou, Christophe Chipot, Wensheng Cai

Overcoming Sampling Issues and Improving the Computational Efficiency in Collective-Variable-Based Enhanced Sampling Simulations

Haohao Fu[*,†,‡,§], *Mengchen Zhou*[†,‡], *Christophe Chipot*[¶,#,⊥] *Wensheng Cai*[*,†,‡,§]

[†]Research Center for Analytical Sciences, Tianjin Key Laboratory of Biosensing and Molecular Recognition, State Key Laboratory of Medicinal Chemical Biology, College of Chemistry, Nankai University, Tianjin 300071, China

[‡]Haihe Laboratory of Sustainable Chemical Transformations, Tianjin 300192, China

[§]School of Materials Science and Engineering, Smart Sensing Interdisciplinary Science Center, Nankai University, Tianjin 300350, China

[¶]Laboratoire International Associé CNRS and University of Illinois at Urbana−Champaign, UMR n°7019, Université de Lorraine, BP 70239, F-54506 Vandœuvre-lès-Nancy, France

[#]Department of Physics, University of Illinois at Urbana−Champaign, 1110 West Green Street, Urbana, Illinois 61801, United States

[⊥]Department of Biochemistry and Molecular Biology, University of Chicago, Chicago 60637, United States

ABSTRACT

This tutorial is designed to help users overcome sampling challenges and improve computational efficiency in collective-variable (CV)-based enhanced sampling enhanced sampling or importance sampling? simulations In addition, remember that when use as an adjective, compound nouns are hyphenated. Toward this end, we introduce well-tempered metadynamics-extended adaptive biasing force (WTM-eABF) and its integration with Gaussian accelerated molecular dynamics (GaMD) as the primary algorithms does it mean that there will be other, secondary algorithms?. Additionally, ~~we present~~ use will be made of a method for identifying the rigorously correct I would be careful with this strong statement least-free-energy pathway (LFEP) on high-dimensional free-energy surfaces For instance, is this method able to identify multiple, concurrent pathways?. We illustrate these sampling techniques~~, we use~~ with the reversible conformational ~~changes~~ transition I personally prefer to use "conformational equilibria" of trialanine and chignolin in aqueous solution as test cases ~~introductory examples~~. This tutorial assumes that the user has prior experience with molecular dynamics (MD) simulations, in general, and with the popular program NAMD, and to some extent with Colvars, the module for CV-based calculations. This tutorial can, however, ~~most of the tutorial is also applicable~~ in large measure be used in conjunction with alternate ~~to other~~ MD engines that support the Colvars module, such as GROMACS, LAMMPS, and Tinker-HP.

Background and Theory

Enhanced sampling algorithms have become essential tools for investigating ~~(bio)~~chemical get rid of the "bio", because if it is biochemical, it is first and foremost chemical processes that are not ~~accessible within~~ amenable to the typical timescale of brute-force molecular dynamics (MD) simulations performed on common computer architectures.[1–4] Among these algorithms, those based on collective variables (CV)~~-based~~ ~~methods~~, also known as importance-sampling methods there seems to be an enormous confusion in the vocabulary. There are importance-sampling schemes that are not based on CVs, have been particularly successful in doing what?.[5] This family of algorithms includes ~~techniques such as~~ umbrella sampling (US),[6] adaptive biasing force (ABF),[7] metadynamics (MtD),[8] temperature-accelerated molecular dynamics (TAMD),[9] as well as their numerous variants.[10–16] These methods an algorithm is not a method leverage a reaction-coordinate model composed of a set of CVs, along which biasing forces or potentials are applied to facilitate sampling along the direction of the CVs. You may choose, like Tony Lelièvre, to distinguish between global algorithms (ABP) and local algorithms (ABF).

Despite their ~~utility~~ undisputed usefulness, CV-based enhanced-sampling Please, correct hyphenation everywhere needed algorithms can ~~encounter~~ face significant challenges when applied to complex processes, which are rooted in two common issues, namely~~-arise:~~ (i) the exploration of the CV space may be slow, and (ii) the exploration of the CV space may appear adequate, but the resulting free-energy surface is not reasonable too vague, too imprecise.[17,18]

Under most circumstances, these issues stem from a common already used in the last sentence problem—the selected set of CVs may not effectively capture the slow degrees of freedom underlying the ~~(bio)~~chemical process.[17] In other words, free-energy barriers in the space orthogonal to that ~~characterized by~~ of the chosen CVs—often referred to as the orthogonal space—can ~~obstruct~~ hamper efficient sampling along the latter.[19]

Improving the selection of CVs, most commonly based on physical intuition, is typically the first step to address these challenges. However, ~~it is frequently the case that researchers' chemical and geometrical intuition fails~~ physical intuition regularly proves insufficient to produce suitable sets of CVs, due to the inherent complexity of the ~~(bio)~~chemical processes at hand, and the practical limitation this limitation needs to be explained on the number of CVs (usually ≤ 3) used in enhanced sampling simulations.[5] For instance, ~~despite~~ notwithstanding extensive studies,[20–23] there is still no universally accepted reaction-coordinate model for describing the reversible folding of proteins.

To ~~enhance their ergodicity, or sampling efficiency~~ improve ergodic sampling you need to explain this concept in both the CV and orthogonal spaces, the efficiency of enhanced sampling algorithms ~~are~~ is continuously updated. In this tutorial, we introduce two advanced enhanced sampling algorithms, namely well-tempered metadynamics-extended adaptive biasing force (WTM-eABF)[24,25] and its ~~integration~~ combination with Gaussian accelerated molecular dynamics (GaMD).[26,27] The latter algorithm emphasizes orthogonal-space sampling by integrating directly with an orthogonal-sampling algorithm repetitive and poorly phrased and is particularly advantageous when the subset of CVs is incomplete, and, hence, insufficient for capturing the slow degrees of freedom of the molecular assembly. ~~The~~ Detail of WTM-eABF and GaMD-WTM-eABF (GaWTM-eABF) can be found in the original publications.[24–26] In what follows, we provide a brief overview of their underlying theories.

In WTM-eABF, the potential energy can be expressed as:

$$U(\mathbf{s},\mathbf{x}) = U(\mathbf{x}) + \frac{1}{2}k[\mathbf{s} - \mathbf{z}(\mathbf{x})]^2 + U_\mathrm{b}(\mathbf{s},t)$$

where $U(\mathbf{x})$ represents the force-field energy, $\mathbf{z}(\mathbf{x})$ denotes the set of CVs, and $\mathbf{s}$ are the extended degrees of freedom coupled with $\mathbf{z}(\mathbf{x})$ through fictitious springs. The term $U_\mathrm{b}(\mathbf{s},t)$ is a time-dependent biasing potential that both lowers the free-energy barriers and fills the valleys simultaneously. During the simulation, $U_\mathrm{b}(\mathbf{s},t)$ evolves to accelerate the movement of the extended degrees of freedom, $\mathbf{s}$. Since $\mathbf{s}$ is coupled with the CVs, ~~the~~ sampling along the CVs is significantly enhanced.

In GaWTM-eABF, the sampling in the orthogonal space is explicitly improved by incorporating the GaMD potential. The potential energy in GaWTM-eABF is given by:

$$U(\mathbf{s},\mathbf{x}) = U(\mathbf{x}) + \Delta U(\mathbf{x}) + \frac{1}{2}k[\mathbf{s} - \mathbf{z}(\mathbf{x})]^2 + U_\mathrm{b}(\mathbf{s},t)$$

where $\Delta U(\mathbf{x})$ is the GaMD potential designed to flatten the entire potential energy surface, ~~regardless~~ irrespective of the specific CVs. You need to reinforce the idea that while the family of ABF algorithms flattens, indeed, the free-energy landscape along the CV, GaMD will scale down primarily the torsional barriers.

In addition to CV-based enhanced sampling simulations supplying free-energy surfaces as a result, we ~~demonstrate~~ you don't demonstrate anything show how to determine the least-free-energy pathway (LFEP) using the multidimensional lowest energy pathway finder (MULE) algorithm.[25] It is crucial for end-users to remember that, although the free-energy differences obtained from well-converged enhanced sampling simulations are reliable, the free-energy barriers may be inaccurate if the ~~chosen~~ model reaction coordinate formed by the subset of CVs ~~are not~~ is suboptimal, i.e., does not provide a faithful account of the dynamics of the molecular process at play.

Prerequisites

The molecular assemblies and the corresponding CVs used in this study are depicted in Figure 1. We assume that the reader is familiarized with molecular modeling. ~~we provide~~ Amber-format input files[28] are provided in both the Supporting Information (SI) and our GitHub repository (github.com/fhh2626/Tutorial_Advanced_WTM-eABF).

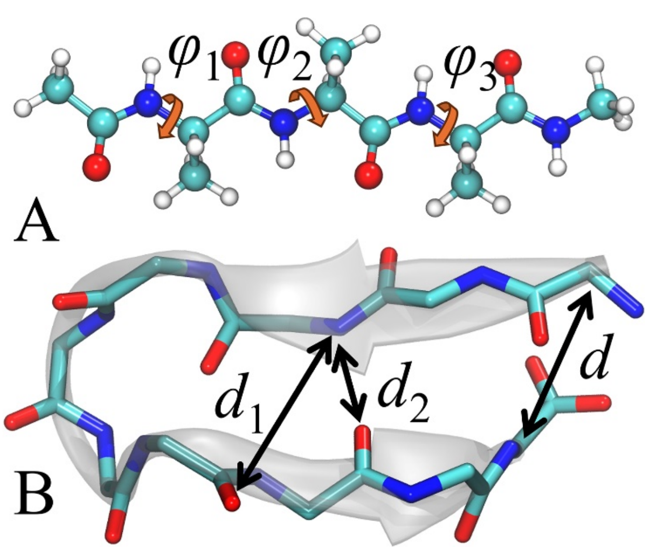

Figure 1. Molecular assemblies used in this tutorial to illustrate the use of CV-based enhanced sampling methods. Trialanine (A) and chignolin (B) with the definition of CVs are shown. For clarity, only the backbone atoms of chignolin are displayed.

This tutorial uses NAMD[29] with its built-in Colvars[30] module as the primary simulation engine. We further assume that users have a prior experience with MD simulations employing NAMD. At the time of writing, the latest version of NAMD is 3.0b7. We recommend to always use the latest version of NAMD to follow this tutorial. A guide for downloading and installing the latest version of NAMD, as well as submitting jobs, is provided in the SI. It should be noted that most of the tutorial is also applicable to other MD engines that support the Colvars module, such as GROMACS,[31] LAMMPS,[32] and Tinker-HP.[33] If there are significant changes in NAMD or Colvars that necessitate modifications of this tutorial, updates will be provided in our GitHub repository (https://github.com/fhh2626/Tutorial_Advanced_WTM-eABF).

For visualization of the results, any MD visualization and scientific plotting software can be used. In this tutorial, Python—along with the Numpy,[34] Scipy,[35] Matplotlib,[36] and Mayavi[37] libraries—is used for analysis and scientific plotting. MULE[25] is utilized to find the LFEP on the multidimensional free-energy surface. PMFToolBox[38] is employed for reweighting the free-energy surfaces. Instructions for installing these tools are provided in the SI.

The tutorial is designed to minimize computational effort. Any computer with a medium-end graphics processing unit (GPU), such as an NVIDIA GeForce RTX 4060 Ti, can handle the simulations proposed herein within a reasonable wall-clock time. While we assume that the end-user is working on a Linux system, Windows aficionados can also follow this tutorial by utilizing the Windows Subsystem for Linux (WSL).

Exercises

Exercise 1 we need a title

In this first exercise, we perform a three-dimensional free-energy calculation to characterize the conformational ~~change~~ equilibrium of alanine tripeptide and identify the LFEP connecting two metastable states. The primary objective of this exercise is to ~~demonstrate~~ unless I am mistaken, you are again not demonstrating anything. A demonstration would either require a mathematical proof, or a comparison with other schemes showcase the efficiency of WTM-eABF in sampling the CV space and to show that WTM-eABF simulations converge in near-equilibrium conditions unclear; can you, please, elaborate?.

The present exercise requires five files, all supplied in the SI: `trialanine_water.parm7`, `trialanine_water.pdb`, `001_eq.conf`, `002_free_energy.conf` and `002_free_energy.in`. We assume that the reader has a prior experience with MD simulations using NAMD already said verbatim above. The first step consists in running an equilibration simulation using the `001_eq.conf` configuration file.

Once the equilibration is complete, the enhanced sampling simulation can begin. The peptide sequence consists of three L-alanine amino acids are they blocked at each end?. The three backbone dihedrals, namely φ_1, φ_2 and φ_3 (Figure 1A), are believed to characterize the slow degrees of freedom reference!!! They are more than believed. In this exercise, these three dihedrals are selected as CVs to compose the reaction-coordinate model.

First, the three dihedrals need to be defined to inform the MD engine of the set of CVs. In the NAMD configuration file (`002_free_energy.conf`), the following statements initialize the Colvars module using `002_free_energy.in`:

```
colvars                 on
colvarsConfig           002_free_energy.in
```

In `002_free_energy.in`, the section

```
colvar {
    width 5.0
    extendedLagrangian    on
    extendedFluctuation   5.0
    subtractAppliedForce  on
    name phi_1
    dihedral {
        group1 {atomNumbers {5}}
        group2 {atomNumbers {7}}
        group3 {atomNumbers {9}}
        group4 {atomNumbers {15}}
    }
}
```

defines a CV. This CV characterizes a dihedral angle between four atoms, namely atoms 5, 7, 9, and 15, and it is named `phi_1` in the Colvars namespace. The range of this CV is not defined explicitly and defaults to [-180°, 180°], with a bin width of 5°. The range and bin width of CVs directly affect the enhanced sampling simulation; the former specifies the interval within which sampling along the CVs is enhanced, while the latter influences the precision of the simulation.

The following settings define an extended degree of freedom coupled to this CV by means of a fictitious spring. With this definition, biases will be applied to the extended particle instead of the real CV, as discussed in the Background and Theory section. A series of parameters determine the spring constant, mass of the extended particle, and the Langevin friction coefficient for the extended degree of freedom (the reader is referred to the Colvars documentation a reference is needed for further detail). For practical purposes, it is generally good practice to set `extendedFluctuation` equal to `width` and `subtractAppliedForce` to `on`, while leaving the other parameters set to their default values. I shall dispute this. Width is the upper limit. Oftentimes, one needs to use a lower value.

```
extendedLagrangian    on
extendedFluctuation   5.0
subtractAppliedForce  on
```

After defining the CVs, biases can be added. The following section applies two biases—ABF and well-tempered MtD —to the extended CVs `phi_1`, `phi_2` and `phi_3`, as prescribed by the WTM-eABF algorithm:

```
abf {
    colvars             phi_1 phi_2 phi_3
    fullSamples         100
    writeCZARwindowFile on
    historyFreq         5000000
}
metadynamics {
    colvars             phi_1 phi_2 phi_3
    hillWidth           3.0
    hillWeight          0.05
    wellTempered        on
    biasTemperature     4000
}
```

Detail of the meaning of these settings can be found in the Colvars documentation reference needed. Most of the values used in this tutorial are suitable for a broad range of applications. Specifically, `fullSamples` controls the number of preliminary samples required per bin prior to application of the biasing force ~~is introduced~~. This value should be increased (e.g., to 10,000) when orthogonal sampling is anticipated to be significant reference to the Miao et al paper. increasing `fullSamples` will substantially augment the computational cost. `HistoryFreq` defines the frequency of writing the current free-energy surfaces to a file. These intermediate free-energy surfaces can be used to monitor the convergence of the simulation, as will be discussed below. The settings in the metadynamics (MtD) section—namely `hillWidth`, `hillWeight`, and `biasTemperature`—control the width and height of the Gaussians added by MtD, and the bias temperature for the well-tempered strategy,[39] respectively.

Once the enhanced sampling simulation is configured using `002_free_energy.conf` and `002_free_energy.in`, it can be carried out. Achieving convergence in the free-energy calculation may require more than 600 nanoseconds of simulation time, which could take between one to four days of wall-clock time, depending on the GPU model ~~of the GPU used~~.

The main result of the simulation is the free-energy surface, which is saved in the file `free_energy.abf1.czar.pmf`. This free-energy surface can be visualized by plotting a contour map using any scientific plotting software. In this tutorial, we use Python and the Mayavi library for three-dimensional scientific data visualization references needed. Detailed instructions for installing Python and the required libraries are provided in the SI. To visualize the free-energy surface, the user can execute the command python the Python command? The Python script? `3D_plot.py` in PowerShell or in a terminal. This script can be easily modified to adapt to other three-dimensional free-energy surfaces. The output window is shown in Figure 2A, where the user can interactively change the viewpoint. From Figure 2A and the data in `free_energy.abf1.czar.pmf`, the user can determine the CVs characterizing the metastable states. Subsequently, the user can identify which frames correspond to these CVs by examining `free_energy.colvars.traj`, and extract the corresponding structures from the trajectory file `free_energy.dcd` (Figure 2B).

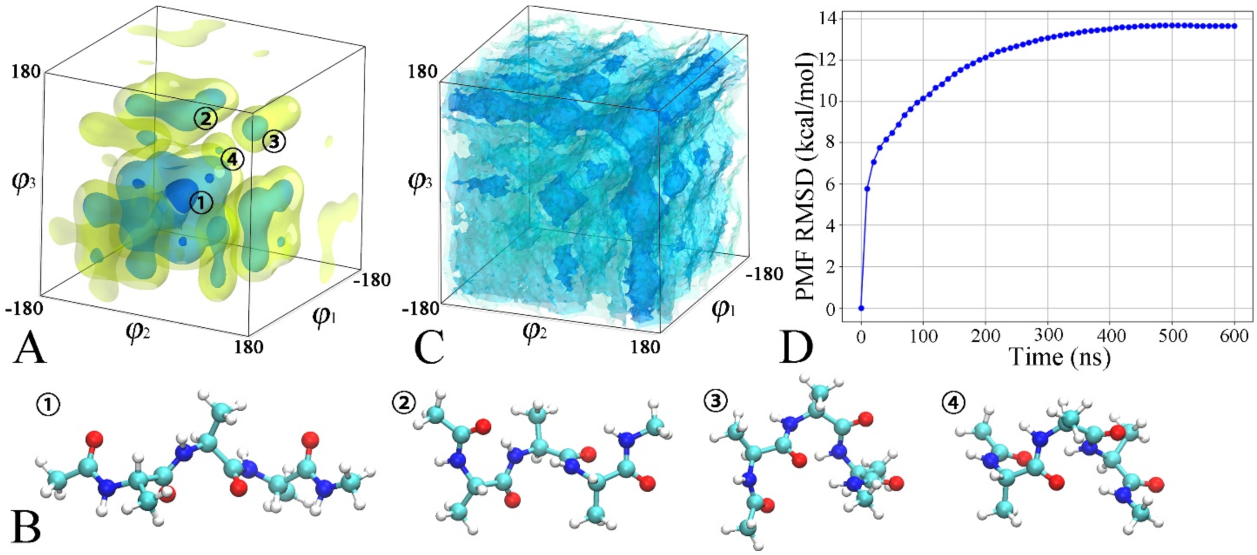

Figure 2. Free-energy surface characterizing the conformational change of trialanine. Blue, cyan, and yellow surfaces correspond to ΔG values of 1.2, 4, and 7 kcal/mol, respectively (A). Corresponding structures of local minima (B). Sampling of the CV space. Blue and cyan surfaces correspond to samples of 1000 and 500, respectively (C). Time evolution of PMF RMSD (D).

To probe the convergence of the simulation, we first examine `free_energy.abf1.zcount` to see if the sampling across the entire CV space is nearly uniform. For easier visualization, a modified version of `3D_plot.py` can be employed. We provide a modified script named `3D_plot_count.py` to plot a three-dimensional contour that highlights well-sampled regions in the CV space. As shown in Figure 2C, no under-sampled region is found in the CV space, and we conclude that sampling is nearly uniform.

In addition, `free_energy.abf1.hist.czar.pmf` saves the free-energy surfaces at different time marks. Convergence of the enhanced sampling simulation can also be validated by plotting the time series of the root-mean-square deviation (RMSD) between the free-energy surfaces and zero unclear and poorly phrased. A script named `pmf_rmsd.py` is provided to achieve this goal. As depicted in Figure 2D, after an initial period of logarithmic increase, the ~~system~~ curve reaches a plateau. This plateau indicates that the output free-energy surface has stabilized over time. ~~This phenomenon~~ It also suggests that the enhanced sampling simulation, or free-energy calculation, has likely converged in a near-equilibrium condition I do not get it. What does this mean?.

All the input files necessary for performing the free-energy calculation using the plain ABF algorithm are provided. The end-user can follow a similar workflow to carry out a plain ABF simulation. The results of these simulations are included in the SI. By comparing the results obtained from ABF (Figure S1) and WTM-eABF simulations, it is evident that ABF cannot ~~thoroughly~~ explore exhaustively the CV space within 600 nanoseconds. In contrast, WTM-eABF significantly ~~enhances~~ improves the sampling efficiency and convergence rate by introducing extended degrees of freedom and metadynamics-like time-dependent biases.

Exercise 2 we need a title

In this exercise, we ~~demonstrate~~ examine how to find the lowest free-energy path (LFEP) on a multi-dimensional free-energy surface. To find the LFEP connecting the global minimum at (–70°, –70°, –70°), and a local stable state at (50°, 50°, 60°) on the free-energy surface saved in `free_energy.abf1.czar.pmf`, a MULE configuration file, `lfep_config.ini`, is supplied:

```
[mule]
directory =    free_energy.abf1.czar.pmf
initial   =   -70,-70,-70
end       =    50, 50, 60
pbc       =     1,  1,  1
```

The `pbc` parameters indicate whether the CVs are periodic. Other settings are self-explanatory. After executing `./mule ./lfep_config.ini` (or `./mule.exe ./lfep_config.ini` on Windows), two output files are generated: `free_energy.abf1.czar.traj` and `free_energy.abf1.czar.energy`. The former file contains the LFEP, while the latter records the free-energy change along the LFEP. Typically, users may want to plot the LFEP as a scatter plot, and the free-energy surface as a contour map on the same figure. To achieve this goal, the script `3D_plot_scatter.py` is provided in the SI. By executing python `3D_plot_scatter.py`, the end-user can generate the plot shown in Figure 3A. Additionally, the free-energy change along the path can be visualized using *s*, the path CV, where *s* varies between 0 and 1, the initial and end states of the transition, respectively. This step can be achieved with the script `energy.py` (Figure 3B).

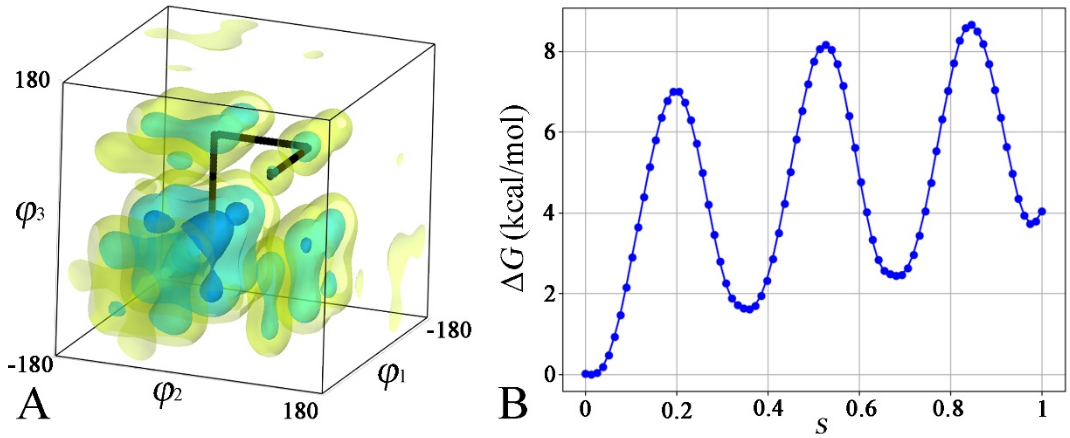

Figure 3. LFEP found by MULE on the three-dimensional free-energy surface characterizing the conformational change of trialanine. Blue, cyan, and yellow surfaces correspond to ΔG values of 1.2, 4, and 7 kcal/mol, respectively (A). Free-energy change along the LFEP (B).

MULE can also be used to find the second and third LFEPs, although the detail of this procedure falls beyond the scope of this tutorial. For those interested in exploring these advanced features, the GitHub repository provides a detailed documentation and several examples of MULE, which the reader may find helpful. Additionally, PMFToolBox, a graphical user interface (GUI) based tool,[38] can be employed to find the LFEP of a given free-energy surface in a more user-friendly fashion. However, MULE offers more flexible features for handling periodic CVs and adding geometric? restraints.

Exercise 3 we need a title

In this exercise, we ~~will~~ investigate the reversible folding of the mini-protein chignolin (Figure 1B) references needed. The end-to-end distance, *d*, is used as the CV for enhanced sampling. While this reaction-coordinate model can evidently not capture all the slow degrees of freedom involved in the ~~protein~~ reversible folding process, ~~it may be challenging to~~ finding a better alternative is admittedly challenging. Therefore, in this case, we turn to the GaWTM-eABF algorithm with this suboptimal reaction-coordinate model to enhance sampling for ~~movement modes~~ collective motions that the selected CV cannot adequately describe.

Unlike dihedral angles, the range and boundaries of other types of CVs are usually explicitly defined, as shown below:

```
lowerboundary                   3.0
upperboundary                   22.0
width                           0.1

reflectingLowerboundary         on
reflectingUpperboundary         on
expandBoundaries                on
```

In `002_free_energy.conf`, the following settings indicate the use of GaMD to improve orthogonal-space sampling:

```
accelMD                         on
accelMDG                        on
accelMDGcMDSteps                3000000
accelMDGcMDPrepSteps            900000
accelMDGEquiPrepSteps           900000
accelMDGEquiSteps               3000000
accelMDdihe                     on
accelMDOutFreq                  5000
accelMDGSigma0D                 6.0
```

GaMD simulations consist of two stages. The first stage, comprising `accelMDGcMDSteps` and `accelMDGEquiSteps`, is for collecting data and estimating the GaMD boost potential. After this stage, the production simulation is performed. The `accelMDdihe` parameter indicates that the boost potential applies to the dihedral energy term. The `accelMDGSigma0D` parameter, set to 6.0, determines the aggressiveness spoken English; we need a better word here of the GaMD boost potential, a value ~~that works well~~ acceptable in most cases. It should be noted that the GPU-resident version of NAMD is currently not compatible with GaMD, so that `CUDASOAIntegrate` should be set to `off` (the default setting) in the ~~NAMD~~ configuration file. As a result, the simulation speed of GaMD-WTM-eABF simulations is lower than that of WTM-eABF. As a corollary, if one is concerned about simulation cost-effectiveness, the former algorithm should be employed only when the selection of CVs is particularly suboptimal.

The following section in `002_free_energy.conf` details the two-stage ~~process of~~ GaWTM-eABF simulation. In the first—preparation—stage, a plain WTM-eABF simulation is performed to collect data and estimate the GaMD boost potential. The second stage corresponds to the actual GaWTM-eABF simulation.

```
for {set stage 0} {$stage < 2} {incr stage} {
    if {$stage == 0} {
        puts "Probing the GaMD parameters..."
        cv configfile 002_free_energy_prepare.in
        run norepeat  6000000
    } elseif {$stage == 1} {
        puts "Starting eABF + GaMD..."
        cv reset
        cv configfile 002_free_energy.in
        run norepeat 1000000000
    }
}
```

In these two stages, different Colvars configuration files are used: `002_free_energy_prepare.in` and `002_free_energy.in`. The bulk of these files is the same, but the latter contains a specific section:

```
reweightamd {
    colvars         end_to_end_distance
}
```

This section instructs the simulation to output an additional file, `free_energy.reweightamd1.cumulant.pmf`, which records the contribution from GaMD to the output free-energy surface. The true free-energy surface along the CV is then the sum of `free_energy.abf1.czar.pmf` and `free_energy.reweightamd1.cumulant.pmf`. The definition of the grids is also slightly different between the two files. To merge the two free-energy surfaces rigorously after interpolation, the script `merge_pmf.py` can be used. After executing `python ./merge_pmf.py ./output/free_energy.abf1.czar.pmf ./output/free_energy.reweightamd1.cumulant.pmf`, the user will obtain `output.pmf`, which can be utilized for further analysis and visualization.

Sometimes, the end-user may want to analyze the free-energy change along a different set of CVs than those used in enhanced sampling here would be a good place to mention Jerome and Giacomo's Dashboard. These ~~new~~ alternate CVs do not need to describe the slow degrees of freedom, but must discriminate between the different metastable states. The (d_1, d_2) coordinates highlighted in Figure 1B can be considered as good candidates for the ~~new~~ alternate CVs.[40] As a prerequisite, ~~these new CVs~~ they must be output in `free_energy.colvars.traj`. This requirement can be achieved by defining them in the Colvars configuration file, as shown in the following example:

```
colvar {
    name asp3n_gly70
    distance {
        group1 {atomNumbers {31}}
        group2 {atomNumbers {92}}
    }
}
```

In `free_energy.colvars.traj`, these CVs are then output in the 3rd and 4th columns (0-based). PMFToolBox can be used to perform the reweighting. As shown in Figure 4A, most options are self-explanatory, but it should be noted that the lower and upper boundaries and the bin? width for the alternate CVs should be set manually after clicking the "Read axes" button. The free-energy surface is then output after clicking the "Run" button

(Figure 4B), which reveals three metastable states of chignolin in aqueous solution, namely a folded (basin 1), a misfolded (basin 2), and an ensemble of extended (basin 3) states.

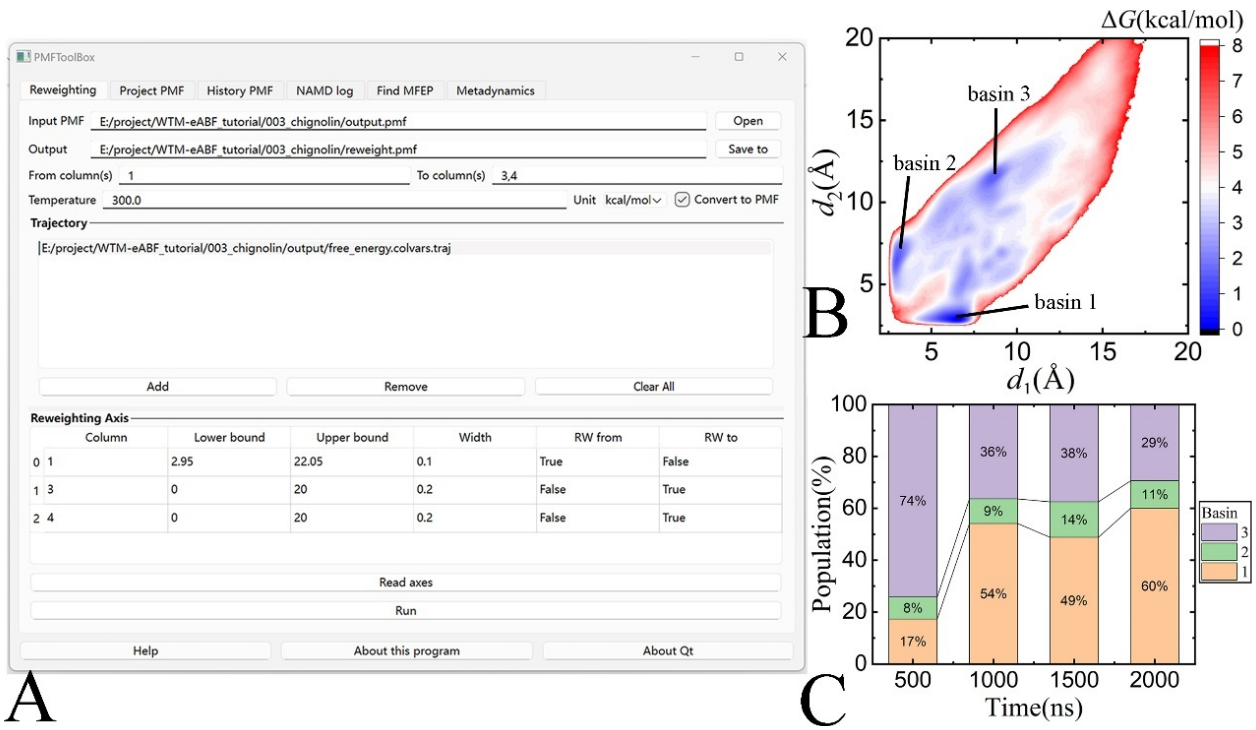

Figure 4. Graphical interface of PMFToolBox and settings for free-energy reweighting of chignolin (A). Free-energy surface along (d_1, d_2) characterizing the reversible folding of chignolin (B). Time series of populations in different basins obtained from the GaWTM-eABF simulation (C).

Since the biases are applied along the end-to-end distance, d, sampling along the new CVs, d_1 and d_2, can be regarded as orthogonal-space sampling. Hence, we demonstrate the orthogonal-space sampling capability poorly phrased. Are we demonstrating anything? We are showing that the algorithm is able to overcome barriers in orthogonal space. of GaWTM-eABF by comparing the exploration of the (d_1, d_2) space with WTM-eABF. This comparison can be directly achieved by counting the samples in `free_energy.colvars.traj`. As shown in Figure 4C, GaWTM-eABF exhibits a significant ability a method does not "exhibit" to sample the orthogonal space compared to WTM-eABF (Figure S2), ~~indicating a reliable result for the obtained free-energy surface from the former~~ there is something wrong with the logic of the sentence. It is important to emphasize that neither d nor (d_1, d_2) ~~well~~ describe appropriately the slow degrees of freedom of the reversible folding of chignolin. The free-energy surface along (d_1, d_2) is only correct in thermodynamics and should be used to distinguish metastable states this sentence is totally opaque. The kinetics, characterized by free-energy barriers on the surface, is not reliable are you saying that important degrees of freedom are missing? Are you saying that markovianity of the variable is questionable? These sentences need to be rewritten.

Conclusion

In this tutorial, through didactic illustrations, we show how to leverage ~~use of~~ two highly efficient CV-based enhanced sampling algorithms ~~methods~~, namely WTM-eABF[24,25] and GaWTM-eABF,[26] using NAMD[29] in conjunction with Colvars.[30] WTM-eABF effectively explores both the CV and orthogonal spaces by combining the advantages of ABF and MtD. GaWTM-eABF further improves orthogonal-space sampling by explicitly integrating GaMD. Since GaWTM-eABF is not supported by the GPU-resident mode of NAMD (though it is supported by the plain GPU mode), its computational efficiency is lower than that of WTM-eABF no. The problem is not the computational efficiency, but the performance. Second, do we really to discuss this? Eventually, everything will be offloaded onto the GPU. Therefore, the end-user should choose the appropriate method for enhanced sampling and free-energy calculation based on their molecular systems and possible sets of CVs to balance convergence rate and computational efficiency. Same comment. In general, we recommend using GaWTM-eABF when the chosen CVs are not expected to describe the slow degrees of freedom too strong (something like "provide a suboptimal description of the slow degrees of freedom") and when the kinetics of the process is not of interest.

In addition, advanced analyses of enhanced sampling simulations, such as finding the LFEP and reweighting, are introduced using tools like MULE[25] and PMFToolBox.[38] Python scripts for plotting and analysis are also provided. These scripts can easily be adapted for simulations of other molecular assemblies at the price of only minor modifications. Aside from NAMD,[29] Colvars is also supported by other MD engines, like GROMACS,[31] LAMMPS,[32] and Tinker-HP.[33] While GaWTM-eABF has not yet been implemented in these programs, the remainder of this tutorial ~~should be applicable in~~ can be handled using these MD programs. We believe ~~this tutorial~~ the present pedagogical material can ~~assist~~ help the community ~~in using~~ exploit advanced enhanced sampling methods to investigate a wide range of ~~(bio)~~chemical processes. Moreover, the CV-based enhanced sampling simulations introduced in this tutorial constitute a prerequisite knowledge essential for geometrical routes[41–43] of absolute binding free-energy calculations. This sentence is poorly phrased, and overly specific for the neophyte. You might want to drop “geometrical route” unless you explain it properly.

ASSOCIATED CONTENT

The Supporting Information is available free of charge at

https://pubs.acs.org/doi/xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

Guide for the installation of software used in this tutorial, results for investigating the conformational change of trialanine using ABF, and results for investigating the reversible folding of chignolin using WTM-eABF (PDF).

Required tutorial files, and binaries of MULE and PMFToolBox for Windows (ZIP).

AUTHOR INFORMATION

Corresponding Author

*E-mail: fhh2626@nankai.edu.cn (H.F.)

*E-mail: wscai@nankai.edu.cn (W.C.)

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENT

This study was supported by the National Natural Science Foundation of China (22103041, 22073050, 22174075, 22293030 and 22293032), the National Key R&D Program of China (2022YFA1305200), the Natural Science Foundation of Tianjin (23JCQNJC01420) and the Haihe Laboratory of Sustainable Chemical Transformations.

REFERENCES

(1) Hénin, J.; Lelièvre, T.; Shirts, M. R.; Valsson, O.; Delemotte, L. Enhanced Sampling Methods for Molecular Dynamics Simulations [Article v1.0]. *Living J. Comput. Mol. Sci.* **2022**, *4* (1), 1583.

(2) Yang, Y. I.; Shao, Q.; Zhang, J.; Yang, L.; Gao, Y. Q. Enhanced Sampling in Molecular Dynamics. *J. Chem. Phys.* **2019**, *151* (7), 70902.

(3) Chen, H.; Chipot, C. Enhancing Sampling with Free-Energy Calculations. *Curr. Opin. Struct. Biol.* **2022**, *77*, 102497.

(4) Bussi, G.; Laio, A. Using Metadynamics to Explore Complex Free-Energy Landscapes. *Nat. Rev. Phys.* **2020**, *2* (4), 200–212.

(5) Fu, H.; Bian, H.; Shao, X.; Cai, W. Collective Variable-Based Enhanced Sampling: From Human Learning to Machine Learning. *J. Phys. Chem. Lett.* **2024**, *15* (6), 1774–1783.

(6) Torrie, G. M.; Valleau, J. P. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *J. Comput. Phys.* **1977**, *23* (2), 187–199.

(7) Darve, E.; Pohorille, A. Calculating Free Energies Using Average Force. *J. Chem. Phys.* **2001**, *115* (20), 9169–9183.

(8) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci.* **2002**, *99* (20), 12562–12566.

(9) Maragliano, L.; Vanden-Eijnden, E. A Temperature Accelerated Method for Sampling Free Energy and Determining Reaction Pathways in Rare Events Simulations. *Chem. Phys. Lett.* **2006**, *426* (1), 168–175.

(10) Invernizzi, M.; Parrinello, M. Rethinking Metadynamics: From Bias Potentials to Probability Distributions. *J. Phys. Chem. Lett.* **2020**, *11* (7), 2731–2736.

(11) Fu, H.; Shao, X.; Chipot, C.; Cai, W. Extended Adaptive Biasing Force Algorithm. An On-the-Fly Implementation for Accurate Free-Energy Calculations. *J. Chem. Theory Comput.* **2016**, *12* (8), 3506–3513.

(12) Lesage, A.; Lelièvre, T.; Stoltz, G.; Hénin, J. Smoothed Biasing Forces Yield Unbiased Free Energies with the Extended-System Adaptive Biasing Force Method. *J. Phys. Chem. B* **2017**, *121* (15), 3676–3685.

(13) Invernizzi, M.; Parrinello, M. Exploration vs Convergence Speed in Adaptive-Bias Enhanced Sampling. *J. Chem. Theory Comput.* **2022**, *18* (6), 3988–3996.

(14) Sidky, H.; Whitmer, J. K. Learning Free Energy Landscapes Using Artificial Neural Networks. *J. Chem. Phys.* **2018**, *148* (10), 104111.

(15) Blumer, O.; Reuveni, S.; Hirshberg, B. Combining Stochastic Resetting with Metadynamics to Speed-up Molecular Dynamics Simulations. *Nat. Commun.* **2024**, *15* (1), 240.

(16) Sousa, C. F.; Becker, R. A.; Lehr, C.-M.; Kalinina, O. V; Hub, J. S. Simulated Tempering-Enhanced Umbrella Sampling Improves Convergence of Free Energy Calculations of Drug Membrane Permeation. *J. Chem. Theory Comput.* **2023**, *19* (6), 1898–1907.

(17) Barducci, A.; Bonomi, M.; Parrinello, M. Metadynamics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2011**, *1* (5), 826–843.

(18) Fu, H.; Shao, X.; Cai, W.; Chipot, C. Taming Rugged Free Energy Landscapes Using an Average Force. *Acc. Chem. Res.* **2019**, *52* (11), 3254–3264.

(19) Zheng, L.; Chen, M.; Yang, W. Random Walk in Orthogonal Space to Achieve Efficient Free-Energy Simulation of Complex Systems. *Proc. Natl. Acad. Sci.* **2008**, *105* (51), 20227–20232.

(20) Sittel, F.; Stock, G. Perspective: Identification of Collective Variables and Metastable States of Protein Dynamics. *J. Chem. Phys.* **2018**, *149* (15), 150901.

(21) Pietrucci, F.; Laio, A. A Collective Variable for the Efficient Exploration of Protein Beta-Sheet Structures: Application to SH3 and GB1. *J. Chem. Theory Comput.* **2009**, *5* (9), 2197–2201.

(22) Palazzesi, F.; Valsson, O.; Parrinello, M. Conformational Entropy as Collective Variable for Proteins. *J. Phys. Chem. Lett.* **2017**, *8* (19), 4752–4756.

(23) Noé, F.; De Fabritiis, G.; Clementi, C. Machine Learning for Protein Folding and Dynamics. *Curr. Opin. Struct. Biol.* **2020**, *60*, 77–84.

(24) Fu, H.; Zhang, H.; Chen, H.; Shao, X.; Chipot, C.; Cai, W. Zooming across the Free-Energy Landscape: Shaving Barriers, and Flooding Valleys. *J. Phys. Chem. Lett.* **2018**, *9* (16), 4738–4745.

(25) Fu, H.; Chen, H.; Wang, X.; Chai, H.; Shao, X.; Cai, W.; Chipot, C. Finding an Optimal Pathway on a Multidimensional Free-Energy Landscape. *J. Chem. Inf. Model.* **2020**, *60* (11), 5366–5374.

(26) Chen, H.; Fu, H.; Chipot, C.; Shao, X.; Cai, W. Overcoming Free-Energy Barriers with a Seamless Combination of a Biasing Force and a Collective Variable-Independent Boost Potential. *J. Chem. Theory Comput.* **2021**, *17* (7), 3886–3894.

(27) Miao, Y.; Feher, V. A.; McCammon, J. A. Gaussian Accelerated Molecular Dynamics: Unconstrained Enhanced Sampling and Free Energy Calculation. *J. Chem. Theory Comput.* **2015**, *11* (8), 3584–3595.

(28) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. Ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from Ff99SB. *J. Chem. Theory Comput.* **2015**, *11* (8), 3696–3713.

(29) Phillips, J. C.; Hardy, D. J.; Maia, J. D. C.; Stone, J. E.; Ribeiro, J. V; Bernardi, R. C.; Buch, R.; Fiorin, G.; Hénin, J.; Jiang, W.; et al. Scalable Molecular Dynamics on CPU and GPU Architectures with NAMD. *J. Chem. Phys.* **2020**, *153* (4), 44130.

(30) Fiorin, G.; Klein, M. L.; Hénin, J. Using Collective Variables to Drive Molecular Dynamics Simulations. *Mol. Phys.* **2013**, *111* (22–23), 3345–3362.

(31) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1–2*, 19–25.

(32) Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; et al. LAMMPS - a Flexible Simulation Tool for Particle-Based Materials Modeling at the Atomic, Meso, and Continuum Scales. *Comput. Phys. Commun.* **2022**, *271*, 108171.

(33) Lagardère, L.; Jolly, L.-H.; Lipparini, F.; Aviat, F.; Stamm, B.; Jing, Z. F.; Harger, M.; Torabifard, H.; Cisneros, G. A.; Schnieders, M. J.; et al. Tinker-HP: A Massively Parallel Molecular Dynamics Package for Multiscale Simulations of Large Complex Systems with Advanced Point Dipole Polarizable Force Fields. *Chem. Sci.* **2018**, *9* (4), 956–972.

(34) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362.

(35) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17* (3), 261–272.

(36) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9* (3), 90–95.

(37) Ramachandran, P.; Varoquaux, G. Mayavi: 3D Visualization of Scientific Data. *Comput. Sci. Eng.* **2011**, *13* (2), 40–51.

(38) Chen, H. PMFToolBox. Zenodo 2024.

(39) Barducci, A.; Bussi, G.; Parrinello, M. Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method. *Phys. Rev. Lett.* **2008**, *100* (2), 20603.

(40) Oshima, H.; Re, S.; Sugita, Y. Replica-Exchange Umbrella Sampling Combined with Gaussian Accelerated Molecular Dynamics for Free-Energy Calculation of Biomolecules. *J. Chem. Theory Comput.* **2019**, *15* (10), 5199–5208.

(41) Raniolo, S.; Limongelli, V. Ligand Binding Free-Energy Calculations with Funnel Metadynamics. *Nat. Protoc.* **2020**, *15* (9), 2837–2866.

(42) Fu, H.; Chen, H.; Blazhynska, M.; Goulard Coderc de Lacam, E.; Szczepaniak, F.; Pavlova, A.; Shao, X.; Gumbart, J. C.; Dehez, F.; Roux, B.; et al. Accurate Determination of Protein:Ligand Standard Binding Free Energies from Molecular Dynamics Simulations. *Nat. Protoc.* **2022**, *17* (4), 1114–1141.

(43) Govind Kumar, V.; Polasa, A.; Agrawal, S.; Kumar, T. K. S.; Moradi, M. Binding Affinity Estimation from Restrained Umbrella Sampling Simulations. *Nat. Comput. Sci.* **2023**, *3* (1), 59–70.

TOC Graphic: