

University of North Carolina at Chapel Hill

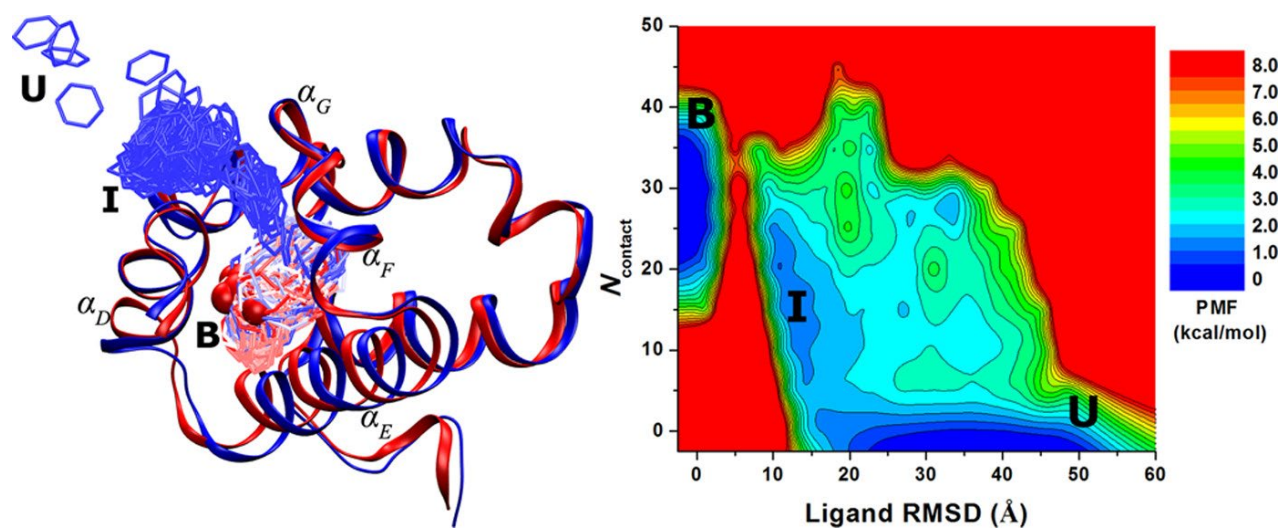
Department of Pharmacology and

Computational Medicine Program

GaMD Workshop 2024

Gaussian accelerated Molecular Dynamics (GaMD) Tutorial

Victor Adediwura, Keya Joshi, Jinan Wang, Amy Liu, and Yinglong Miao



Abstract

Gaussian accelerated Molecular Dynamics (GaMD) is a biomolecular enhanced sampling method that works by adding a harmonic boost potential to smooth the system potential energy surface. This tutorial explains how you can perform GaMD simulations using the Amber24 simulation package. Alanine dipeptide is used as a model system for demonstration. You will learn how to run GaMD simulations to sample multiple conformational states of Alanine dipeptide in explicit solvent and reweight GaMD simulations to calculate the system free energy profiles. Knowledge about standard Molecular Dynamics (MD) simulations is required. Experiences with Amber and understanding of GaMD theory are preferred for learning the tutorial.

Table of contents

1. Introduction	4
2. Run GaMD Simulations on Alanine Dipeptide	7
2.1 System Setup	7
2.2 GaMD Equilibration	8
2.3 GaMD Production Simulations	12
3. Simulation Analysis	14
3.1 Calculate Dihedral Angles	14
3.2 Reweight GaMD Simulations to Calculate Free Energy Profiles	14
4. Questions	21
5. References	21

1. Introduction

Gaussian accelerated Molecular Dynamics (GaMD) is a biomolecular enhanced sampling method that works by adding a harmonic boost potential to smooth the system potential energy surface¹. GaMD greatly reduces energy barriers and accelerates biomolecular simulations by orders of magnitude. GaMD does not require predefined reaction coordinates or collective variables and is thus advantageous for unconstrained enhanced sampling of large biomolecular complexes. Since the GaMD boost potential exhibits a Gaussian distribution, biomolecular free energy profiles can be properly recovered through cumulant expansion to the second order (“Gaussian approximation”). Applications of GaMD have revealed mechanisms of protein folding and conformational changes, ligand binding, protein-protein/membrane/nucleic acid interactions and carbohydrate dynamics²⁻⁴. Please refer to Ref. 1 for details of the GaMD method.

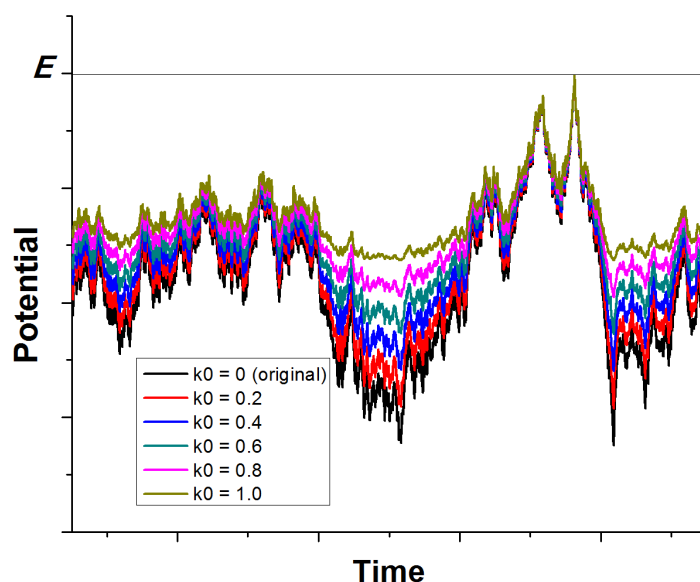


Figure 1: In GaMD, a harmonic boost potential calculated with an effective harmonic force constant k_0 and threshold energy E is added to smooth the potential energy surface.

In this tutorial, we will use alanine dipeptide as a model system to learn the following:

- How to run GaMD simulations with different dihedral-, total- and dual-boost potentials?
- How to analyze simulation trajectories?
- How to reweight GaMD simulations for Potential of Mean Force (PMF) free energy calculations?

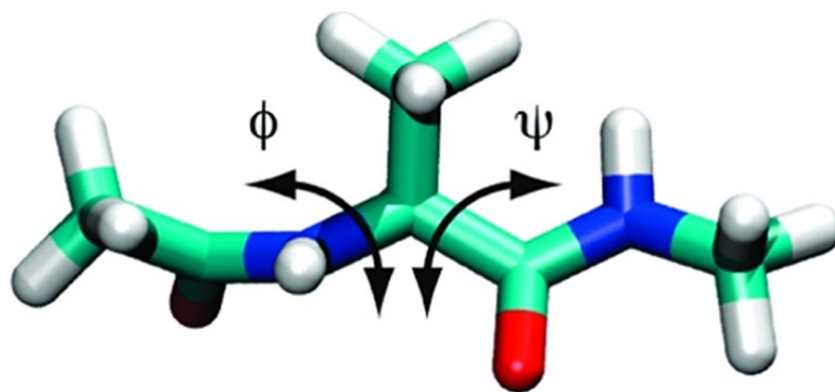


Figure 2. Scheme representation of backbone dihedrals Φ and Ψ in alanine dipeptide.

Required files:

- Files required for the GaMD tutorial can be downloaded at [GaMD.tar.bz2](https://ga-md.org/GaMD.tar.bz2) (24.5MB) or copied from the following folder on the UNC cloud server: `/shared/data/GaMD`. Further instructions on how to obtain these files will be discussed in subsequent sections (**Fig 3**).
- AMBER24+ installed (<https://ambermd.org>)
- Python3 with NumPy and SciPy installed (<https://www.python.org/>)

As shown in **Fig. 3**, the GaMD folder contains two sub-folders: simulation and analysis.

- The **simulation** folder includes eight files: *Ala.parm7* and *cmd.rst7*, representing the system's topology and coordinate files after 10ns cMD, respectively; *md.in* and *gamd-restart.in*, representing the input files for GaMD equilibration and production, respectively; *run-equil.pbs* and *run-prod.pbs*, the scripts used to submit jobs in the UNC cloud via the SLURM queue system; *run-equil.sh* and *run-prod.sh*, the bash scripts to run corresponding GaMD equilibration and production simulations using your own workstations.
- The **analysis** folder includes 17 files: *Ala.pdb* (initial PDB file for visualization), *analysis.in* (input file for *cpptraj*), *gamd-all.log* (the GaMD log file), *Psi_all.dat* (reaction coordinate file calculated with *cpptraj*), *Phi_all.dat* (reaction coordinate file calculated with *cpptraj*), *run-[1-2]d-pmf.pbs* (SLURM submitting scripts for UNC cloud), *run-[1-2]d-pmf.sh* (bash scripts for 1-2D PMF calculations using your own workstation), *reweight-[1-2]d.sh* (bash scripts for reweighting calculations), *PyReweighting-[1-2]D.py* (Python scripts for reweighting calculations), and *create-[1-2]-anharmonicity.sh* (bash scripts for analyzing the boost potential and calculating its anharmonicity).

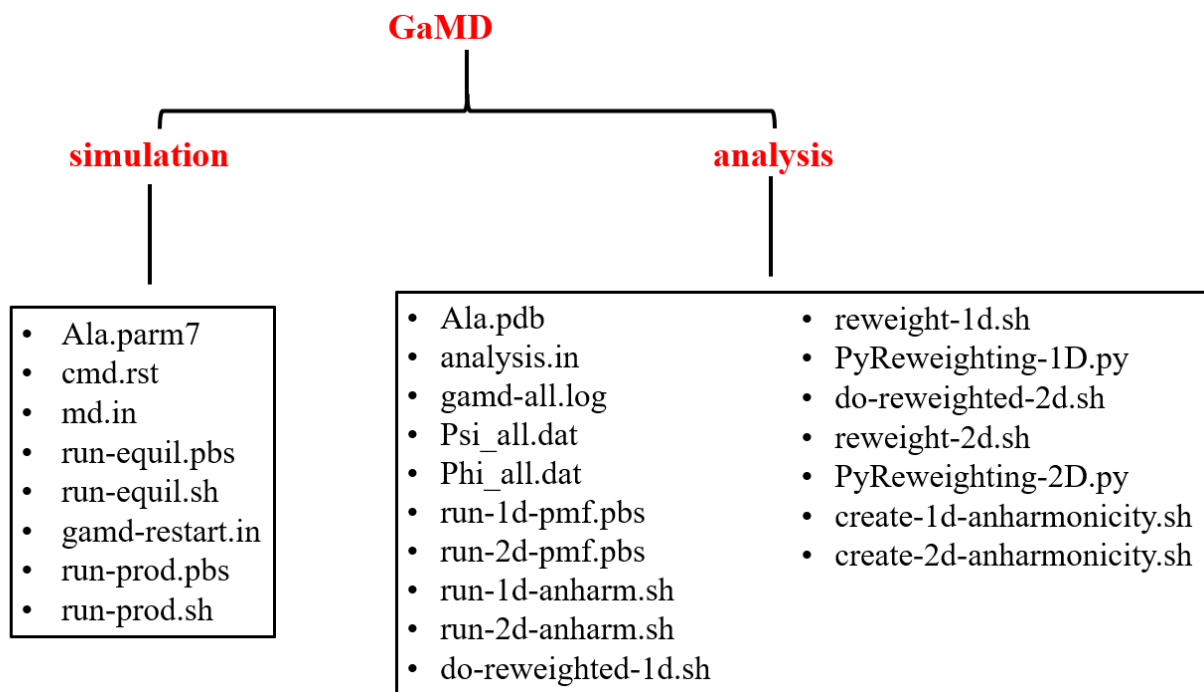


Figure 3: Directory structure of GaMD tutorial files.

• PyReweighting scripts can be downloaded at [PyReweighting24.tar.bz2](#) (12KB) or copied from the following folder on the UNC cloud server: /shared/data/PyReweighting24. As shown in **Fig. 4**, there are three folders: 1d-reweighted, and 2d-reweighted, each corresponding to the calculation of 1D, and 2D. Each folder includes two bash scripts (*do-reweighted-[1-2]d.sh* and *reweight-[1-2]d.sh*), and a Python script (*PyReweighting-[1-2]D.py*) for performing the respective calculations.

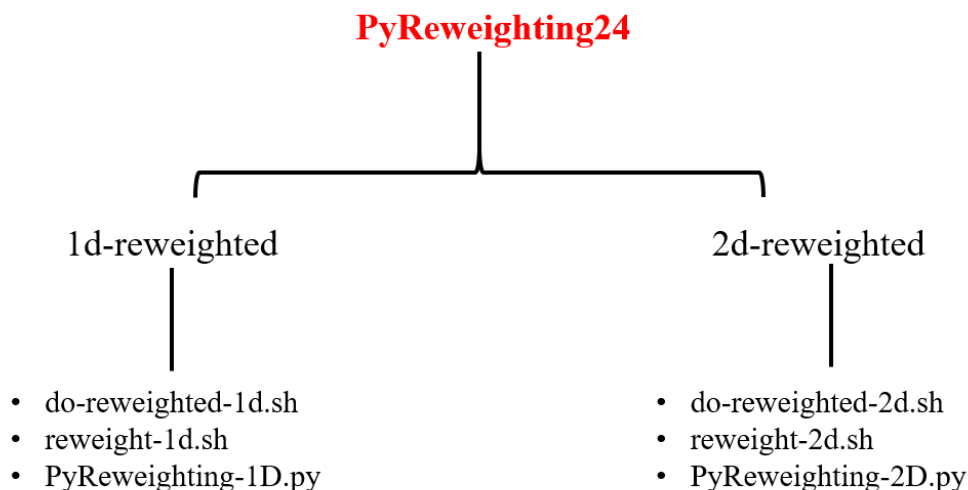


Figure 4: Directory structure of PyReweighting24 files.

2. Run GaMD Simulations on Alanine Dipeptide

2.1 System Setup

Before running GaMD simulation, a short conventional MD (cMD) simulation is needed to obtain a well equilibrated structure. Two recommended approaches for setting up the system are through *tLeap* in AMBER or the CHARMM-GUI website. The following tutorials are highly recommended:

- *tLeap*: <https://ambermd.org/tutorials/pengfei/index.php>
- CHARMM-GUI: <https://ambermd.org/tutorials/CHARMM-GUI.php>

For simplicity, we have prepared the system files after energy minimization, heating, equilibration with restraints, and 10 ns cMD simulation. These files can be found in the “simulation” folder, including the system topology (*Ala.parm7*) and coordinates (*cmd.rst7*). Since cMD is not the focus of this tutorial, we use these files directly.

The following are instructions for using the UNC cloud server:

Make sure that the AMBER environment is properly configured by running the appropriate *source* command. Copy the “GaMD” folder to your home directory:

```
source /shared/apps/software/amber24/amber.sh
cd ~/
cp -r /shared/data/GaMD .
cd ~/GaMD/simulation
```

The following are instructions for users using your own workstation. Make sure that you have downloaded **gamd.tar.bz2** to your home directory and extract the tutorial files:

```
source /shared/apps/software/amber24/amber.sh ## this one needs to be modified according to
your own local system
tar -xvjf gamd.tar.bz2
cd ~/GaMD/simulation
```

In the simulation folder, you should see the following files, *cmd.rst7*, *Ala.parm7*, *run-equil.pbs*, *run-equil.sh*, *run-prod.pbs*, *run-prod.sh*, *md.in*, and *gamd-restart.in*.

2.2 GaMD Equilibration

Run the simulation using pmemd.cuda

In the *run-equil.pbs* file, you will find the command below used to run the GaMD equilibration simulation.

For local/UNC account users:

```
sbatch run-equil.pbs
```

The batch script to run the GaMD equilibration on UNC computing cloud:

```
#!/bin/sh
#SBATCH --job-name=equilibration
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
#SBATCH --gres=gpu:1
#SBATCH --qos=gpu_access
source /shared/apps/software/amber24/amber.sh
which pmemd.cuda
#gamd equilibration
pmemd.cuda -O -i md.in -p Ala.parm7 -c cmd.rst7 -o md-1.out -x md-1.nc -r gamd-1.rst -gamd
gamd-1.log
```

For remote users using your own workstation:

In the *~/GaMD/simulation* directory, run the bash script *run-equil.sh*:

```
cd ~/GaMD/simulation
chmod +x run-equil.sh
./run-equil.sh &
```

A copy of *run-equil.sh* is shown below:

```
#!/bin/sh
#make sure to modify the path of amber24 to the correct location
#source /shared/apps/software/amber24/amber.sh
which pmemd.cuda
# gamd equilibration
pmemd.cuda -O -i md.in -p Ala.parm7 -c cmd.rst7 -o md-1.out -x md-1.nc -r gamd-1.rst -gamd
gamd-1.log
```


You should obtain the following files: *md-1.out*, *md-1.rst*, *md-1.nc*, *gamd-1.log* and *gamd-restart.dat*.

- *md-1.out*: Simulation output file containing the system energies.
- *md-1.nc*: Trajectory file storing the MD simulation frames.
- *gamd-1.rst*: Restart file including both coordinate and velocity information.
- *gamd-1.log*: Logfile of the boost potential values used for reweighting.
- *gamd-restart.dat*: Stores the potential statistics used to apply boosts for the GaMD simulations.
- *mdinfo*: Updates the current state of the simulation, useful for monitoring progress and checking the simulation status.

The *gamd-restart.dat* file is generated in the working directory, which will be used for calculating the parameters to apply boost, *thus it is very important*.

Input file: md.in

GaMD simulation

&cntrl

```
imin=0,      ! No minimization
irest=0,     ! This IS a new MD simulation
ntx=1,       ! read coordinates only
! Temperature control
ntt=3,       ! Langevin dynamics
gamma_ln=1.0, ! Friction coefficient (ps^-1)
tempi=310.0, ! Initial temperature
temp0=310.0, ! Target temperature
ig=-1,      ! random seed
! Potential energy control
cut=9.0,     ! nonbonded cutoff, in Angstroms
! MD settings
nstlim=1500000, ! simulation length was reduced for the sake of time
dt=0.002,    ! time step (ps)
! SHAKE
ntc=2,       ! Constrain bonds containing hydrogen
ntf=2,       ! Do not calculate forces of bonds containing hydrogen
tol=0.000001
! Control how often information is printed
```

```

ntpr=50,    ! Print energies every 1000 steps
ntwx=50,    ! Print coordinates every 1000 steps to the trajectory
ntwr=500,   ! Print a restart file every 10K steps (can be less frequent)
! ntwv=-1,   ! Uncomment to also print velocities to trajectory
! ntwf=-1,   ! Uncomment to also print forces to trajectory
ntxo=2,     ! Write NetCDF format
ioutfm=1,   ! Write NetCDF format (always do this!)
! Wrap coordinates when printing them to the same unit cell
iwrap=1,
ntwprt = 11550,
! Constant pressure control. Note that ntp=3 requires barostat=1
barostat=1, ! MC barostat... change to 1 for Berendsen
ntp=1,      ! 1=isotropic, 2=anisotropic, 3=semi-isotropic w/ surften
pres0=1.0,  ! Target external pressure, in bar
taup=0.5,   ! Berendsen coupling constant (ps)
! Set water atom/residue names for SETTLE recognition
!watnam='TIP3', ! Water residues are named TIP3
!owtnm='OH2', ! Water oxygens are named OH2
! GaMD parameters
igamd = 3, iE = 1, irest_gamd = 0,
ntcmd = 1000000, nteb = 2000000, ntave = 50000,
ntcmdprep = 200000, ntebprep = 200000,
sigma0P = 6.0, sigma0D = 6.0,
/

```

NB: To run a total-boost GaMD simulation, keep all the other parameters in the *md.in* file the same, but set ***igamd* = 1**. If you are interested in a dihedral-boost GaMD simulation, set ***igamd* = 2**.

GaMD input parameters:

igamd Flag to apply boost potential

- = **0** (default) no boost is applied
- = **1** boost on the total potential energy only
- = **2** boost on the dihedral energy only
- = **3** dual boost on both dihedral and total potential energy

irest_gamd Flag to restart GaMD simulation

- = **0** (default) new simulation
- = **1** restart simulation

iE Flag to set the threshold energy E

= **1** (default) set threshold energy to lower bound $E=V_{\max}$

= **2** set threshold energy to upper bound $E=V_{\min}+(V_{\max}-V_{\min})/k_0$

ntcmd Number of CMD steps to calculate V_{\max} , V_{\min} , V_{avg} , σ_V

(default 1,000,000)

nteb Number of biasing equilibration steps (default 1,000,000)

sigma0P Upper limit of the standard deviation of total potential boost (default 6.0 kcal/mol).

sigma0D Upper limit of the standard deviation of dihedral potential boost (default 6.0 kcal/mol).

Example calculations for setting the GaMD simulation steps:

```

natoms= 20 // the number of protein atoms in the system.
natoms_max= 11550 // the total number of atoms in the system
ntwx= 100 // the steps to write the trajectory (default)
ntwprt= natoms= 20 // the number of protein atoms
igamd= 3 (default) // gamd
ntave= 4 x natoms_max = 4 x 11550 = 46,200 = 50,000 (Always make it a whole number)
ntcmd= 10 x ntave = 10 x 50,000 = 500,000
ntcmdprep= 2 x ntave = 2 x 50,000 = 100,000
ntebprep= 2 x ntave = 2 x 50,000 = 100,000
nteb= 40 x ntave = 40 x 50,000 = 2,000,000
nstlim= nteb + ntcmd= 2,000,000 + 500,000 = 2,500,000/500,000 = 5ns

```

NB: The **gamd-restart.dat** should contain the boost potential statistics like this:

-37118.481058826255	-38635.542541253555	-37233.603459717102	27.059672998685546
21.547508955001831	6.0826854947954416	11.645935508938841	1.9239790935750309

To know if the job has been properly submitted, enter:

squeue

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	ODELIST(Reason)
136	hpc	equilibr	vadediwu	CF	0:03	1	RC-GPU-hpc-1

2.3 GaMD Production Simulations

For local/UNC account users: Simply submit the job by entering the command below:

```
sbatch run-prod.pbs
```

A copy of the *run-prod.pbs* batch script:

```
#!/bin/sh
#SBATCH --job-name=production
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
#SBATCH --gres=gpu:1
#SBATCH --qos=gpu_access
source /shared/apps/software/amber24/amber.sh
which pmemd.cuda
pmemd.cuda -O -i gamd-restart.in -p Ala.parm7 -c gamd-1.rst7 -o md-2.out -x md-1.nc -r gamd-1.rst -gamd gamd-1.log
```

For remote users using your own workstation:

```
cd ~/GaMD/simulation
chmod +x run-prod.sh
./run-prod.sh &
```

A copy of the *run-prod.sh* batch script:

```
#!/bin/sh
#make sure to modify the path of amber24 to the correct location
#source /shared/apps/software/amber24/amber.sh
which pmemd.cuda
# gamd equilibration
pmemd.cuda -O -i gamd-restart.in -p Ala.parm7 -c gamd-1.rst -o md-2.out -x md-2.nc -r gamd-2.rst -gamd gamd-2.log
```

Key parameters in the **gamd-restart.in** file include:

- **`irest = 0`**: Indicates a new simulation.
- **`ntx = 1`**: Reads coordinates and velocities from the restart file.
- **`irest_gamd = 1`**: Restarts GaMD parameters.

Here is the complete **gamd-restart.in** file:

GaMD simulation

```
&cntrl
  imin=0,      ! No minimization
  irest=0,     ! This is a new MD simulation
  ntx=1,       ! read coordinates only
  ! Temperature control
  ntt=3,       ! Langevin dynamics
  gamma_ln=1.0, ! Friction coefficient (ps^-1)
  tempi=310.0,  ! Initial temperature
  temp0=310.0, ! Target temperature
  ig=-1,       ! random seed
  ! Potential energy control
  cut=9.0,     ! nonbonded cutoff, in Angstroms
  ! MD settings
  nstlim=15000000, ! simulation length
  dt=0.002,    ! time step (ps)
  ! SHAKE
  ntc=2,       ! Constrain bonds containing hydrogen
  ntf=2,       ! Do not calculate forces of bonds containing hydrogen
  tol=0.000001
  ! Control how often information is printed
  ntp=100,     ! Print energies every 1000 steps
  ntwx=100,    ! Print coordinates every 1000 steps to the trajectory
  ntwr=5000,   ! Print a restart file every 10K steps (can be less frequent)
! ntwv=-1,     ! Uncomment to also print velocities to trajectory
! ntwf=-1,     ! Uncomment to also print forces to trajectory
  ntxo=2,      ! Write NetCDF format
  ioutfm=1,    ! Write NetCDF format (always do this!)
  ! Wrap coordinates when printing them to the same unit cell
  iwrap=1,
  ntwprt = 11550,
  ! Constant pressure control. Note that ntp=3 requires barostat=1
  barostat=1,  ! MC barostat... change to 1 for Berendsen
  ntp=1,       ! 1=isotropic, 2=anisotropic, 3=semi-isotropic w/ surften
  pres0=1.0,   ! Target external pressure, in bar
  taup=0.5,    ! Berendsen coupling constant (ps)
  ! Set water atom/residue names for SETTLE recognition
  !watnam='TIP3', ! Water residues are named TIP3
  !owtnm='OH2',  ! Water oxygens are named OH2
  ! GaMD parameters
```

```
igamd = 3, iE = 1, irest_gamd = 1,  
ntcmd = 0, nteb = 0, ntave = 50000,  
ntcmdprep = 0, ntebprep = 0,  
sigma0P = 6.0, sigma0D = 6.0,
```

/

To check whether the job is correctly submitted and running, enter the command below:

```
squeue
```

3. Simulation Analysis

3.1 Calculate dihedral angles in alanine dipeptide

cd ~/GaMD/analysis

We have already used cpptraj to calculate the dihedral angles and you should see those files named *Psi_all.dat* and *Phi_all.dat*. Additionally, we have the combined GaMD log file named *gamd-all.log*. Double check the analysis folder to ensure that you have those files.

Here, we provide a sample cpptraj script to calculate the dihedral angles, but you don't need to run this script as we have generated the relevant results for you in the *~/GaMD/analysis* folder.

```
parm ~/GaMD/analysis/Ala.parm7  
reference ~/GaMD/analysis/Ala.pdb  
trajin /shared/data/trajectory/gamd.nc 1 last 1 #if you have trajectory files, modify the path  
dihedral Phi :1@C :2@N :2@CA :2@C out Phi_all.dat  
dihedral Psi :1@N :1@CA :1@C :2@N out Psi_all.dat
```

3.2 Reweight GaMD Simulations to Calculate Free Energy Profiles

Because the boost potential exhibits Gaussian distribution in GaMD simulations, cumulant expansion to 2nd order (also referred to as “Gaussian Approximation”) can be effectively used to approximate the ensemble-averaged reweighting factor:

$$\langle e^{\beta \Delta F} \rangle = \exp \left\{ \sum_{k=1}^{\infty} \frac{\beta^k}{k!} C_k \right\},$$

where the first two order cumulants are given by:

$$C_1 = \langle \Delta V \rangle,$$

$$C_2 = \langle \Delta V^2 \rangle - \langle \Delta V \rangle^2 = \sigma_{\Delta V}^2.$$

The Exponential Average and Maclaurin series expansion reweighting results are normally less accurate, but they are also made available in *PyReweighting* for comparison.

To characterize the extent to which ΔV follows Gaussian distribution, its distribution anharmonicity γ is calculated:

$$\gamma = S_{\max} - S_{\Delta V} = \frac{1}{2} \ln(2\pi e \sigma_{\Delta V}^2) + \int_0^\infty p(\Delta V) \ln(p(\Delta V)) d\Delta V,$$

where ΔV is dimensionless as divided by $k_B T$ with k_B and T being the Boltzmann constant and system temperature, respectively, and $S_{\max} = \frac{1}{2} \ln(2\pi e \sigma_{\Delta V}^2)$ is the maximum entropy of ΔV . When γ is zero, ΔV follows exact Gaussian distribution with sufficient sampling. Reweighting by approximating the exponential average term with cumulant expansion to the second order is able to accurately recover the original free energy landscape. As γ increases, the ΔV distribution becomes less harmonic and the reweighted free energy profile obtained from cumulant expansion to the second order would deviate from the original.

To calculate 1D PMF free energy profiles, the following files are needed:

Pyrewighting-1D.py

do-reweighted-1d.sh

reweight-1d.sh

Note: the total number of lines in gamd-all.log should be the same as that of the reaction coordinate file (e.g., Psi_all.dat). To check it, enter this command:

```
wc -l gamd-all.log Psi_all.dat
```

For local/UNC account users:

```
sbatch run-1d-pmf.pbs
```

A copy of the *run-1d-pmf.pbs* script:

```
#!/bin/sh
#SBATCH --job-name=pmf-1d
```

```
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
source /shared/apps/software/amber24/amber.sh
cd ~/GaMD/analysis
chmod +x do-reweighted-1d.sh
./do-reweighted-1d.sh Psi_all.dat gamd-all.log
```

For remote users using your own workstation:

```
./run-1d-pmf.sh &
```

A copy of the *run-1d-pmf.sh* bash script:

```
#!/bin/sh
cd ~/GaMD/analysis
chmod +x do-reweighted-1d.sh
./do-reweighted-1d.sh Psi_all.dat gamd-all.log
```

The reweighted PMF output files can be found in folder **pmf-Psi_all-bin6.0-cutoff300** as shown below:

```
cd pmf-Psi_all-bin6.0-cutoff300/
ls -lrth
```

You will then get the files listed below:

```
pmf-c3-Phi.dat-reweight-disc6.0.dat.xvg
pmf-c2-Phi.dat-reweight-disc6.0.dat.xvg
pmf-c1-Phi.dat-reweight-disc6.0.dat.xvg
pmf-Phi.dat-noweight-disc6.0.dat.xvg
```

The file **pmf-c2-Phi.dat-reweight-disc6.0.dat.xvg** is the reweighted 1D PMF.

To calculate 2D PMF profiles of (Phi, Psi), the following files are needed:

```
Pyreweighting-2D.py
do-reweighted-2d.sh
reweight-2d.sh
```


For local/UNC account users:

```
sbatch run-2d-pmf.pbs
```

A copy of the *run-2d-pmf.pbs* script:

```
#!/bin/sh
#SBATCH --job-name=pmf-2d
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
source /shared/apps/software/amber24/amber.sh
cd ~/GaMD/analysis
chmod +x do-reweighted-2d.sh
./do-reweighted-2d.sh Phi_all.dat Psi_all.dat gamd-all.log
```

For remote users:

```
./run-2d-pmf.sh &
```

A copy of the *run-2d-pmf.sh* bash script:

```
#!/bin/sh
cd ~/GaMD/analysis
chmod +x do-reweighted-2d.sh
./do-reweighted-2d.sh Psi_all.dat Phi_all.dat gamd-all.log
```

The 2D PMF results can be found in the folder **pmf-2d-binx10.0-biny10.0-cutoff300**:

```
cd pmf-2d-binx10.0-biny10.0-cutoff300  
ls -lrth
```

You will see a file named *pmf-2D-c2-Phi_Psi-reweight-discx10.0-discy10.0.xvg*. This is the file that contains the reweighted 2D PMF.

Next, we will analyze the GaMD boost potential distribution and anharmonicity in 1D:

For UNC account users, submit the job by:

```
sbatch run-1d-anharm.pbs
```

Here is a copy of the *run-1d-anharm.pbs* script:

```
#!/bin/sh
#SBATCH --job-name=1d-anharmonicity
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
#SBATCH --gres=gpu:1
#SBATCH --qos=gpu_access
source /shared/apps/software/amber24/amber.sh
./create-1d-anharmonicity.sh Psi_all.dat gamd-all.log
```

For remote users using your own workstation:

```
./run-1d-anharm.sh &
```

A copy of the *run-1d-anharm.sh* bash script:

```
#!/bin/bash
if [ $# != 2 ]
then
    echo "usage example: ./do-reweighted-1d.sh Psi_all.dat gamd-all.log"
    exit
fi
RC1=$1
logfile=$2
TEMPERATURE=310
wc -l ${RC1} ${logfile}
awk '{print $2}' ${RC1} | tail -n +1 > psi.dat
paste -d "\t" psi.dat >> Psi
grep -v "#" ~/GaMD/analysis/gamd-all.log > headerless-gamd.log
awk 'NR%1==0' headerless-gamd.log | awk -v TEMP=TEMPERATURE -v
IDEAL_GAS_CONSTANT=0.001987 '{print ($8+$7)/(IDEAL_GAS_CONSTANT*TEMP)}' "$2"
" ($8+$7)}' > weights.dat
python3 PyReweighting-1D.py -input Psi -T TEMPERATURE -cutoff 300 -Xdim -180 180 -disc
6 -Emax 100 -job amd_dV -weight weights.dat | tee -a reweight-variable-cumulant-expansion-
1D.log
```

In the *~/GaMD/analysis* folder, *dV-hist-Psi.xvg* and *dV-anharm-Psi.xvg* are the boost potential distribution and the anharmonicity.

We can also analyze the GaMD boost potential distribution and anharmonicity in 2D as follows.

For UNC account users, submit the job by:

```
sbatch run-2d-anharm.pbs
```

A copy of the *run-2d-anharm.pbs* script:

```
#!/bin/sh
#SBATCH --job-name=2d-anharmonicity
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:20:00
#SBATCH --output=log.%x.%j
#SBATCH --gres=gpu:1
#SBATCH --qos=gpu_access
source /shared/apps/software/amber24/amber.sh
./create-2d-anharmonicity.sh Phi_all.dat Psi_all.dat gamd-all.log
```

For remote users using your own workstation:

```
./run-2d-anharm.sh &
```

Software tools like `xmgrace`, `gnuplot`, or `Originpro` can be used to plot the results.

- 1D Free Energy Profile:

xmgrace can be used to plot the *pmf-psi-reweight-CE2.xvg*

```
xmgrace dV-hist-Psi.dat.xvg
```

-2D Free Energy Surface:

You should get something like the following if you use the Originpro!

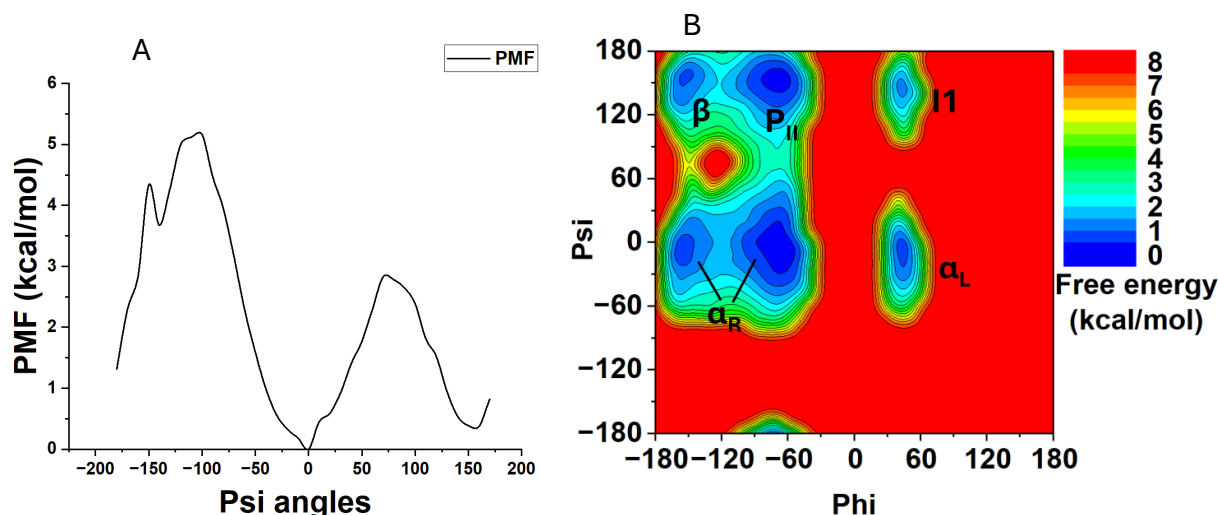


Figure 5: (A) 1D potential mean force (PMF) of the Psi dihedral angle in alanine dipeptide. (B) 2D PMF of backbone dihedrals phi (Φ), psi (Ψ) calculated from three 30 ns GaMD simulations combined using cumulant expansion to the second order. The low-energy wells are labeled corresponding to the right-handed α helix (α_R), left-handed α helix (α_L), β -sheet (β), and polyproline II (P II) conformations. The low energy state II could be an intermediate state.

Examine the 1D and 2D anharmonicity:

Again, xmgrace can be used to plot the *dV-anharm-Psi.dat.xvg* and Originpro can be used to plot the 2D anharmonicity (*dV-anharm-2D-Phi_Psi.xvg*):

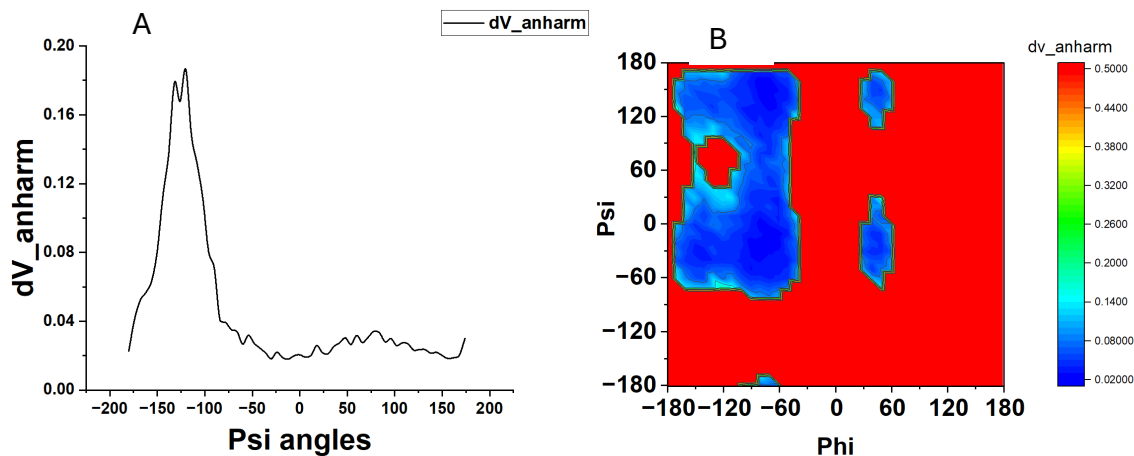


Figure 6: Distribution anharmonicity of ΔV of frames found in each bin of the (A) 1D and (B) 2D PMF profiles.

Examine the 1D and 2D boost potential distribution:

Xmgrace can be used to plot the *dV-hist-Psi.dat.xvg* & *dV-hist-2D-Phi_Psi.xvg*:

xmgrace dV-hist-Psi.dat.xvg

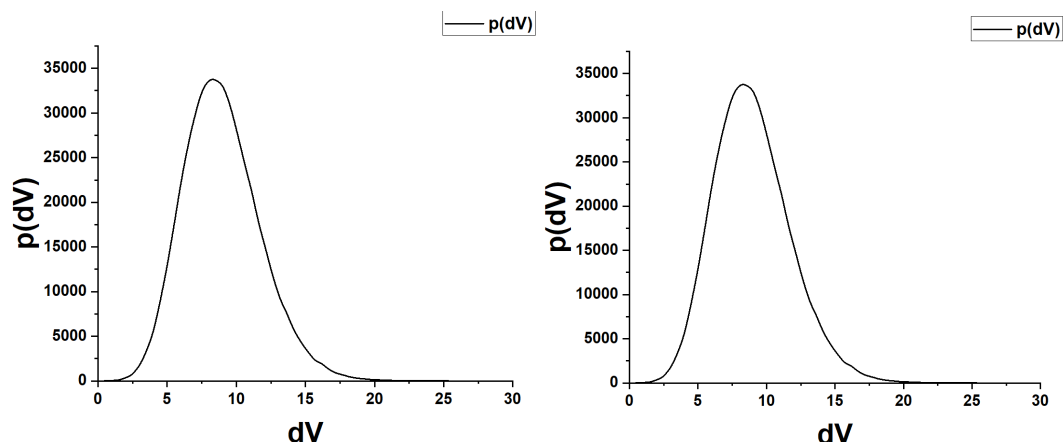


Figure 7: Distribution of the boost potential ΔV applied in the GaMD simulation

4. Questions

If you encounter any issues or have further questions, please post them to GaMD mailing list: gamd-discuss@lists.sourceforge.net after subscription at <https://sourceforge.net/projects/gamd/lists/gamd-discuss>. For updates and latest tutorials of GaMD, visit the GaMD website: <https://www.med.unc.edu/pharm/miaolab/resources/gamd/>.

5. References

- 1 Miao, Y., Feher, V. A. & McCammon, J. A. Gaussian Accelerated Molecular Dynamics: Unconstrained Enhanced Sampling and Free Energy Calculation. *J. Chem. Theory Comput.* **11**, 3584-3595 (2015). <https://doi.org/10.1021/acs.jctc.5b00436>
- 2 Wang, J. *et al.* Gaussian accelerated molecular dynamics: principles and applications. *WIREs Comput. Mol. Sci.*, e1521 (2021). <https://doi.org/10.1002/wcms.1521>
- 3 Miao, Y. & McCammon, J. A. Gaussian Accelerated Molecular Dynamics: Theory, Implementation and Applications. *Annual Reports in Computational Chemistry* **13**, 231-278 (2017). <https://doi.org/10.1016/bs.arcc.2017.06.005>
- 4 Bhattarai, A. & Miao, Y. Gaussian Accelerated Molecular Dynamics for Elucidation of Drug Pathways. *Expert Opinion on Drug Discovery* **13**, 1055-1065 (2018). <https://doi.org/10.1080/17460441.2018.1538207>