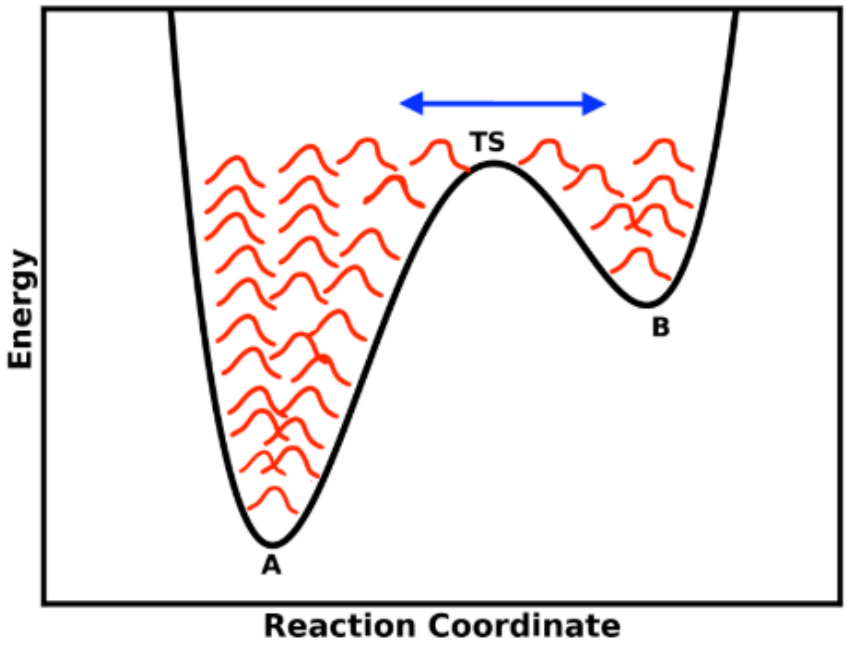
Practical 3: Metadynamics

Introduction

Molecular dynamics simulations are useful to determine all-atomistic detailed information about how a chemical system moves. Yet propagating Newton's equations of motions for a simple system such as a medium-sized protein in solution can be very computationally expensive due to it containing hundreds of thousands of atoms. Proteins undergo dynamics under a wide range of timescales. Hydrogen bound fluctuations occur at femtosecond timescales, while large-scale conformational changes occur in range of microseconds to seconds. The large-scale conformational changes of proteins are typically what biologists are interested in because those motions determine biologically important functions such as for example whether a ligand can bind to the binding-pocket of the protein.

Using regular simulations it is not possible to reach the timescales of these biologically relevant events. Large conformational changes in proteins have high free energy barriers which are difficult to escape from. To observe these large conformational changes we can add artificial energy to the system to try to push the system over the free energy barrier. The artificial energy being added in metadynamics is are mathematical gaussian functions along some collective variable which describes the progress of going from one initial state to another final state. As the simulation progresses more and more gaussians are added until the system moves over the free energy barrier.



https://www.plumed-tutorials.org/lessons/22/001/data/plumed_masterclass_22-1.html

But, if you add artificial energy into a natural system you are no longer simulating something realistic. The time of the transition is much faster than what would happen realistically, and the probability distribution is distorted. But, due to some convenient properties of gaussian functions we can correct for the affect of the bias and recover unbiased probabilities from a biased simulation.

Here I show a derivation which shows how to correct for the bias.

$$P(\vec{x}) = \frac{1}{Z} e^{-\frac{U(\vec{x})}{k_B T}}$$
$$Z = \int d\vec{x} e^{-\frac{U(\vec{x})}{k_B T}}$$

$$P_{bias}(\vec{x}) = \frac{1}{Z'} e^{-\frac{U(\vec{x})+V_{bias}}{k_B T}}$$
$$Z' = \int d\vec{x} e^{-\frac{U(\vec{x})+V_{bias}}{k_B T}}$$

$$P_{bias}(\vec{x}) = \frac{1}{Z'} e^{-\frac{U(\vec{x})}{k_B T}} e^{-\frac{V_{bias}}{k_B T}}$$

$$\frac{P(\vec{x})}{P_{bias}(\vec{x})} = \frac{\frac{1}{Z} e^{-\frac{U(\vec{x})}{k_B T}}}{\frac{1}{Z'} e^{-\frac{U(\vec{x})}{k_B T}} e^{-\frac{V_{bias}}{k_B T}}}$$

$$\frac{P(\vec{x})}{P_{bias}(\vec{x})} = \frac{Z'}{Z} e^{\frac{V_{bias}}{k_B T}}$$

$$P(\vec{x}) = \frac{Z'}{Z} P_{bias}(\vec{x}) e^{\frac{V_{bias}}{k_B T}}$$

$$U(\vec{x}) = -k_B T ln\big(P(\vec{x})\big)$$

These equations essentially tell us that if we can set-up a simulation that evolves according to its internal energy $U(\vec{x})$ and a defined bias V_{bias} and observe the probability from this biased simulation $P_{bias}(\vec{r})$, then we can recover the unbiased probability $P(\vec{r})$ and unbiased internal energy $U(\vec{r})$.

But, if we are interested in getting free energy along a reaction-coordinate we have to define a marginal probability.

$$P(s) = \int \delta(s - s(\vec{x}))\, P(\vec{x}) d\vec{x}$$

Substitute in definitions

$$P(s) = \int \delta(s - s(\vec{x})) \frac{Z'}{Z} P_{bias}(\vec{x}) e^{\frac{V_{bias}}{k_B T}} d\vec{x}$$

Let $w(\vec{x}) = \frac{P(\vec{x})}{P_{bias}(\vec{x})} = \frac{Z'}{Z} e^{\frac{V_{bias}}{k_B T}}$

$$P(s) = \int \delta(s - s(\vec{x})) \, P_{bias}(\vec{x}) w(\vec{x}) d\vec{x}$$

This can be approximated with a summation

$$P(s) = \frac{1}{N} \sum_{k=1}^{N} \delta(s - s(\vec{x}_k)) w(\vec{x}_k)$$

So, when we sum over the points in the biased probability distribution and correct by a weight due to the bias, then we can get the unbiased probability with respect to the collective variable.

Then, from that expression we can get the unbiased free energy with respect to the collective variable.

$$F(s) = -k_B T ln\big(P(s)\big)$$

Setting up Metadynamics Simulation Starting from NPT Outputs

Metadynamics is a variation to running a production run. A production run is an unbiased simulation. While metadynamics is an enhanced sampling method. The inputs for the metadynamics simulation are the outputs of the npt equilibration step.

If you started setting up an unbiased simulation as we have been doing in this class, then you need the following inputs

pro.mdp
npt_{prot_name}.gro
sys.top

These inputs will be used to create a pro_{prot_name}.tpr which contains the information for how the production simulation will be run.

To create the pro_{prot_name}.tpr run the following commands

```
module load gromacs/2024.1

gmx_mpi grompp -f pro.mdp    -c npt_${prot_name}.gro    -r npt_${prot_name}.gro    -o
pro_${prot_name}.tpr    -p ./sys.top -maxwarn 5
```

This step will be useful to start metadynamics simulations for your Final Project Structures. We can skip this step for our in class example, since we will just use an already made pro_{prot_name}.tpr following the tutorial PLUMED: Advanced Methods in MD 2023: Metadynamics simulations with PLUMED. The simple system being studied in this tutorial is alanine dipeptide in vacuum.

Run Unbiased Production Run of Alanine Dipeptide in Vacuum

First, we need to run an unbiased production run simulation for the alanine dipeptide in vacuum and track the collective variable that we are interested in. We know that alanine dipeptide has a large free energy barrier along its phi dihedral angle. We will track the value of the phi dihedral angle as the simulation progresses. Later on we will use the phi dihedral angle as a collective variable to bias the system.

First copy the folders "alanine_dipeptide_unbiased_simulation_for_ch447_547" and "alanine_dipeptide_metadynamics_simulation_for_ch447_547" from the "ch447_547/shared" folder into your folder in "ch447_547".

When running the unbiased simulation codes you should be in the unbiased simulation folder. When running the metadynamics simulation codes you should be in the metadynamics simulation folder.

To run an unbiased production run of alanine dipeptide in vacuum use the following command…

sbatch run_plumed_unbiased_MD.sh

This will cause the following script to run.

```
#!/bin/bash
#SBATCH --partition=compute      ### queue to submit to
#SBATCH --job-name=run_plumed_sim        ### job name
#SBATCH --output=run_plumed_sim_jobout         ### file in which to store job stdout
#SBATCH --error=run_plumed_sim_joberr          ### file in which to store job stderr
#SBATCH --time=0-00:30:00        ### Wall clock time limit in HH:MM:SS
#SBATCH --nodes=2               ### number of nodes to use
#SBATCH --ntasks-per-node=24    ### nuimber of tasks to launch per node
#SBATCH --cpus-per-task=1       ### number of cores for each task
#SBATCH --mail-user=abatela2@uoregon.edu
#SBATCH --mail-type=begin  # email me when the job starts
#SBATCH --mail-type=end    # email me when the job finishes
#SBATCH --account=guenzagrp      ### Account used for job submission

module purge
module load gcc/13.1.0
module load openmpi/4.1.6
module load gromacs/2024.3-plumed-2.11.0

#export PLUMED_KERNEL="/gpfs/packages/plumed/2.9.2/src/lib/libplumedKernel.so"

gmx_mpi mdrun -s topol.tpr -nsteps 5000000 -plumed plumed_unbiased.dat
```

The gromacs command in this script contains an input plumed_unbiased.dat file which tells the script how to keep track of the phi dihedral angle.

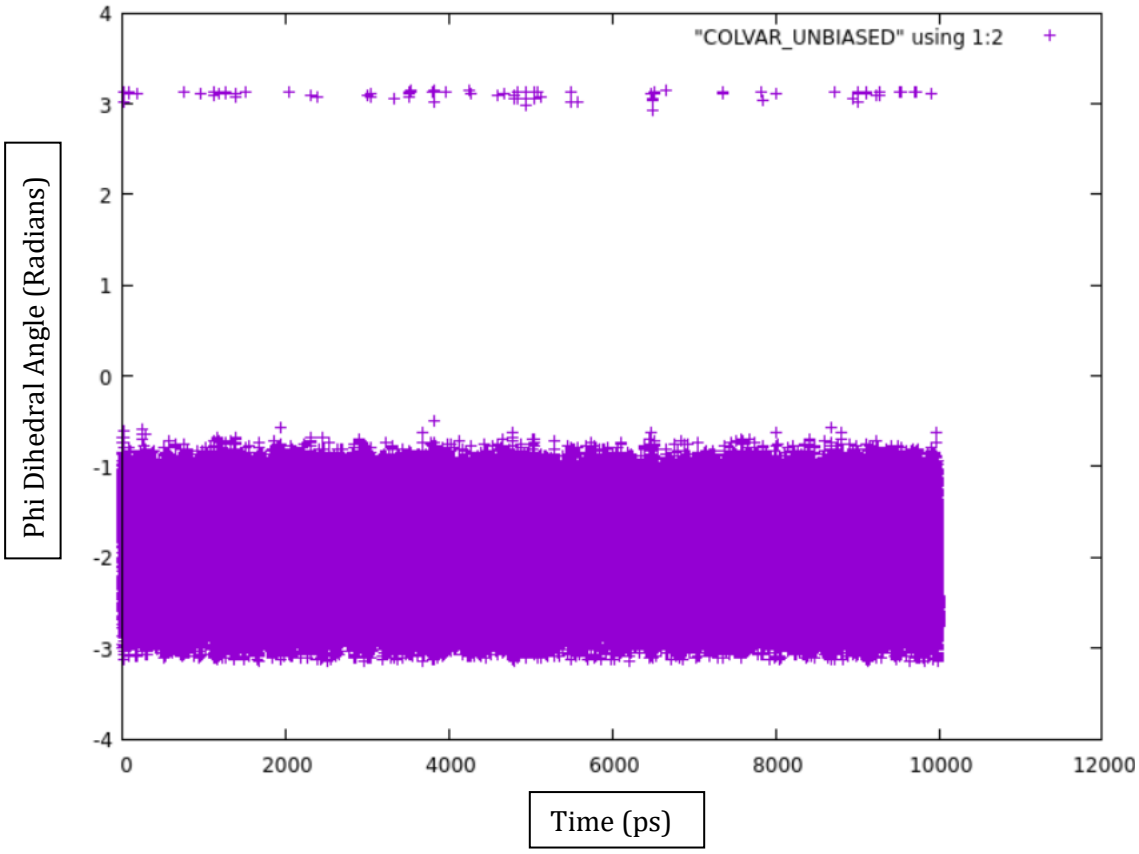
The plumed_unbiased.dat file contains the following information

```
# vim:ft=plumed

MOLINFO STRUCTURE=diala.pdb
phi: TORSION ATOMS=5,7,9,15
psi: TORSION ATOMS=7,9,15,17

# Print both collective variables on COLVAR file every 10 steps
PRINT ARG=* FILE=COLVAR_UNBIASED STRIDE=10

#ENDPLUMED
```

This takes the pdb structure of alanine dipeptide as input and calculates the dihedrals phi and psi using the plumed TORSION command. The atoms specified for each dihedral are the 4 points necessary to describe a dihedral angle. Then, it prints out the dihedral angles as a function of time into the output file COLVAR_UNBIASED.

To plot the phi dihedral angle vs time type in the command

gnuplot
set datafile commentschars "#"
plot "COLVAR_UNBIASED" using 1:2 with points



We can see that in the unbiased simulation that the system stays in a region of phi dihedral angle of around -1 to -3 radians and never visits the region of dihedral angle of around 0-2 radians. We could have started the simulation in the other minima. Then, we would see the system only stay in that minima.

Run Metadynamics Production Run of Alanine Dipeptide in Vacuum

To run a metadynamics production run of alanine dipeptide in vacuum use the following command…

sbatch run_plumed_MD.sh

This will cause the following script to run.

```
#!/bin/bash
#SBATCH --partition=compute      ### queue to submit to
#SBATCH --job-name=run_plumed_sim        ### job name
#SBATCH --output=run_plumed_sim_jobout         ### file in which to store job stdout
#SBATCH --error=run_plumed_sim_joberr          ### file in which to store job stderr
#SBATCH --time=0-00:30:00        ### Wall clock time limit in HH:MM:SS
#SBATCH --nodes=2                ### number of nodes to use
#SBATCH --ntasks-per-node=24     ### nuimber of tasks to launch per node
#SBATCH --cpus-per-task=1        ### number of cores for each task
#SBATCH --mail-user=abatela2@uoregon.edu
#SBATCH --mail-type=begin  # email me when the job starts
#SBATCH --mail-type=end    # email me when the job finishes
#SBATCH --account=guenzagrp      ### Account used for job submission

module purge
module load gcc/13.1.0
module load openmpi/4.1.6
module load gromacs/2024.3-plumed-2.11.0

#export PLUMED_KERNEL="/gpfs/packages/plumed/2.9.2/src/lib/libplumedKernel.so"

gmx_mpi mdrun -s topol.tpr -nsteps 5000000 -plumed plumed.dat
```

The gromacs command in this script contains an input plumed.dat file which tells the script how to keep track of the phi dihedral angle and also biases the system along the phi dihedral angle.

The plumed.dat file contains the following information

```
# vim:ft=plumed

MOLINFO STRUCTURE=diala.pdb
phi: TORSION ATOMS=5,7,9,15
psi: TORSION ATOMS=7,9,15,17

metad: METAD ...
  ARG=phi
  PACE=500
  HEIGHT=1.2
  SIGMA=0.1
  BIASFACTOR=10
  # Gaussians will be written to file and also stored on grid
  FILE=HILLS GRID_MIN=-pi GRID_MAX=pi
...

...

# Print both collective variables on COLVAR file every 10 steps
PRINT ARG=* FILE=COLVAR STRIDE=10

#PRINT FMT=%g STRIDE=500 FILE=COLVAR ARG=*

#ENDPLUMED
```

This structure again load the alanine dipeptide pdb structure file. It calculates the dihedral angles using the plumed TORSION command.

But, now the new part comes in.

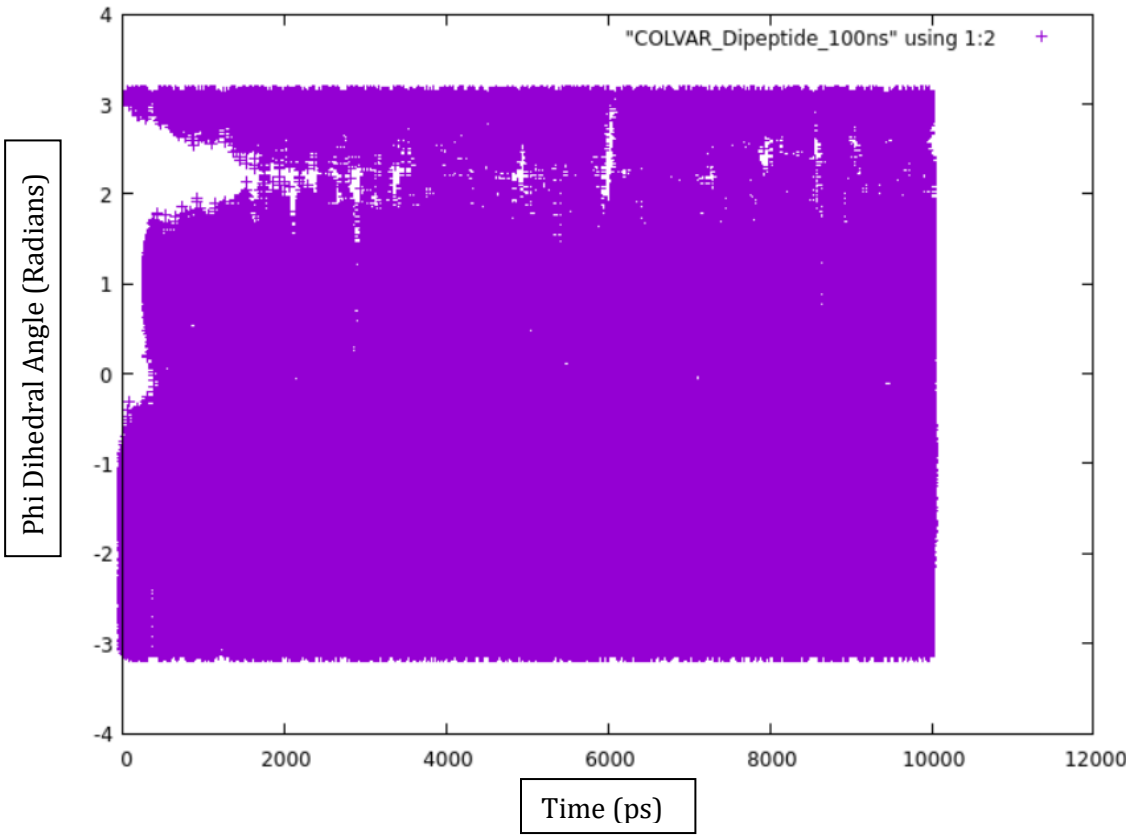
We run metadynamics using the metad command. This command takes multiple inputs which specify how metadynamics will be run.

ARG is which collective variable we will bias.
PACE is how quickly the gaussians are placed into the simulation.
HEIGHT is the height of each guassian.
SIGMA is the width of each gaussian.
BIASFACTOR causes the gaussian height to decrease over time to allow the free energy to converge.
FILE input specifies that we will save bias information to a file called HILLS and store the bias on a grid. The grid helps save computational space, since tracking values on a grid in low-dimensions is more efficient than keeping track of each individual gaussian.

Finally, we print all the outputs including dihedral angles to an output file called COLVAR.

To plot the phi dihedral angle vs time for the metadynamics simulation type in the command

```
gnuplot
set datafile commentschars "#"
plot "COLVAR" using 1:2 with points
```



We can see that in the metadynamics simulation that the system transitions between both minima now. The system is no longer stuck in any minima and is exploring the full -pi to pi radians of possible dihedral angle space. But, if we would try to now look at the probability distribution for this metadynamics simulation it is biased. So, if we wouldn't do a correction than the relative probability for being in each minima would not be correct.

In the next tutorial we will go over how to obtain the unbiased probability distribution and unbiased free energy from these biased metadynamics simulation results.