Part 0: General Remarks

GROMACS Version: For this assignment, uses GROMACS version 2023.1 . There is no restriction as to how the GROMACS executable is compiled, so long as it is the correct version. The executable used in this competition **must be** `gmx` (GROMACS simulations using multiple nodes simultaneously are not allowed in this competition).

Restricted Flags for mdrun: The use of the -nsteps and -maxh flags is forbidden. For any trajectory to count in this competition, it must be run to completion for the amount of time specified in the initial mdp or tpr file. Modifications of the provided mdp files are also forbidden.

File Submission Specifications: For this final assignment, output files must be returned in a tar.gz archive called `GROMACS_<your-team-name>_submission.tar.gz` , where <your-team-name> is replaced with the name of your team. The instructions for naming and organization of the submitted files must be followed **exactly.** Due to storage considerations, **please do not submit additional files beyond what is requested.** The details of the names and organization of the files will be listed in the instructions for each part of this assignment. Additionally, for your convenience, a sample submission tar.gz file named `example_submission.tar.gz` is provided that contains directory structures and empty files with the proper file names. You can use the structure of the provided .tar.gz file as a guide for your submission.

In addition to submitting a .tar.gz file, you will also need to fill out the spreadsheet `GROMACS_calculated_values.ods`, rename it as `GROMACS_<your-team-name>_calculated_values.ods` and submit separately from the .tar.gz file.

Part 1: Entropy Estimation of a Bound and Unbound c-Met Inhibitor

This section counts for 25% of your final score

Preamble: One of the challenges in developing a compound that can inhibit protein function is that protein-ligand binding is a process that is inherently entropically unfavorable. In this exercise, we will construct protein-ligand and ligand-only systems, run the MD trajectories, and then use the resulting output to calculate the entropy of the ligand in the protein-bound and unbound forms. We will use the c-Met system (4R1Y) for this exercise.

Instructions - MD Simulations: Follow a similar protocol in setting up the protein-ligand and ligand-only systems as we did in the homework. New scripts are provided in the attached files, with the main difference being that the production run (07_md_run.tpr) runs for 5 ns, significantly longer than what was done in the homework.

Pull the 4R1Y crystal structure from the Protein Data Bank. The ligand residue in this case is labeled '3EH'. The non-protein residues are '3EH', '7PE', and 'HOH'. Otherwise, follow the steps for creating a Protein+Ligand system and a Ligand-Only system as detailed in the homework (sections 2.0 - 4.0). Put the protein+ligand system in a folder called "protein/", and the ligand-only system in a folder called "water/". Minimize and equilibrate both systems as you did in the homework. Then, run the 5 ns MD simulations (07_md_run) for both systems however you see fit (you do not need to run them multiple times with different numbers of processors). **For both systems, keep the `topol.top` file and name the final trajectory, log file, and tpr file as `07_md_run.xtc`, `07_md_run.log` and `07_md_run.tpr`, respectively. Resubmit the mdp file that you used as well.** Expect the protein system simulation to require roughly 5 hours of compute time on a single node.

Instructions - Entropy Calculations: With the final simulations in hand, we can use the output trajectories to calculate the total entropy of the bound and unbound ligand. Do this by following these steps for both systems:

1. Create a new directory called `entropy_calculation`, and enter it.
2. Create a trajectory that corrects for periodic boundary conditions, translation, and rotation. Failure to do so may negatively impact your entropy calculation. **Only output the ligand for this trajectory.** Possible names for the ligand selection include 'LIG', 'MOL', or '3EH'. Name this file `entropy-trajectory.xtc`
3. Use the gmx covar command to create covariance matrix and output a non-mass weighted trajectory. Name this file `entropy-trajectory_nomwa.trr`
4. Use the same gmx covar command to create a mass-weighted (with the -mwa option) eigenvector .xvg file. Name this file `entropy-trajectory_mwa.xvg`
5. Use the _nomwa.trr and _mwa.xvg files to calculate entropy at 300 Kelvin using the following gmx anaeig command:

```
gmx anaeig -v entropy-trajectory_nomwa.trr \
 -eig entropy-trajectory_mwa.xvg -entropy -temp 300
```

6. The output of the common in step 5 will give an estimated entropy value using the

Schlitter method. **Report this value in the spreadsheet to be submitted**.

PLEASE NOTE: You will have trouble with steps 2-4 because you will be required to renumber atoms. This will require you to create a new tpr file (gmx convert-tpr), and create a new index file with proper numbering (gmx make_ndx). You can consult the relevant pages of the GROMACS documentation for details on the syntax of these commands. **Please include the final tpr and index files you used for the entropy calculation process**. Name them as `entropy-tpr.tpr` and `entropy-index.ndx` .

Requisite Files for Submission for Part 1:

```
part_1/
   |------protein/
   |          |--------07_md_run.log
   |          |--------07_md_run.mdp
   |          |--------07_md_run.xtc
   |          |--------07_md_run.tpr
   |          |--------index.ndx
   |          |--------topol.top
   |          |--------entropy_calculation/
   |                     |--------entropy-index.ndx
   |                     |--------entropy-tpr.tpr
   |                     |--------entropy-trajectory_mwa.xvg
   |                     |--------entropy-trajectory_nomwa.trr
   |                     |--------entropy-trajectory.xtc
   |------water/
              |--------07_md_run.log
              |--------L07_md_run.mdp
              |--------07_md_run.xtc
              |--------07_md_run.tpr
              |--------index.ndx
              |--------topol.top
              |--------entropy_calculation/
                         |--------entropy-index.ndx
                         |--------entropy-tpr.tpr
                         |--------entropy-trajectory_mwa.xvg
                         |--------entropy-trajectory_nomwa.trr
                         |--------entropy-trajectory.xtc
```

Part 2: Estimating Relative Binding Free Energies Using Non-Equilibrium Switching

This section counts for 60% of your final score

Preamble: A common use of molecular dynamics is to estimate the free energy of binding of a ligand to a protein, which is directly related to measurable biological data such as IC50 values of protein inhibition. While many methods of free energy calculation have been developed over the years, the recent emergence of non-equilibrium switching has been shown to deliver comparable results to existing methodologies at roughly 1/3rd of the computational cost.

In a non-equilibrium switching experiment, the difference in free energy of binding is calculated between two ligands, referred to as a relative binding free energy, or ΔΔG. In the first step of the process, equilibrium MD trajectories of each ligand are obtained in both the protein-bound (‘protein/’) and unbound (‘water/’) states. Then, snapshots of each simulation are taken, and the ligands are slowly alchemically transmuted from one ligand into the other. While alchemical transmutation does not exist in real life, it is possible within a computer simulation. The process of transmuting one ligand into another allows us to estimate quantities such as binding free energies.

Instructions - Alchemical Transformations: In this assignment, you are provided with the tpr files built from the snapshots of different trajectories of 2 protein systems (c-Met and MUP1), and their unbound ligands. In each simulation, a ligand is alchemically transmuted into a different ligand. 4 replicas of each simulation were used, and each simulation was broken up into 64 snapshots, which yields 1024 protein-bound tpr files and 1024 unbound (water) tpr files between the different sets.

Your task is to run all 2048 MD simulations, and then use the data to calculate two ΔΔG values. You will be graded not only on the successful completion of the different stages of this task, **but will also be graded on a scale relative to other teams based on the average ns/day of each of the groups of simulations that are performed**.

The grading based on ns/day is based on four sections, using each of the following directories per section:

c-Met/protein
c-Met/water
MUP1/protein
MUP1/water

As an example, if the average (arithmetic mean) of your team’s c-Met/protein runs os 2.5 ns/day, and the best ns/day among all participants is 3.0 ns/day, then your team would get a score of 83.33% for the c-Met/protein section. Because the scoring is segmented by section, if you are unable to complete all of the MD runs, it is recommended to focus on finishing some sections rather than others.

Below is the directory tree of the c-Met/ directory provided for this stage of the assignment:

```
c-Met/
    |------protein/
    |           |--------stateA/
    |           |           |--------run1/
    |           |           |           |---------nes_{00..63}.tpr
    |           |           |--------run2/
    |           |           |           |---------nes_{00..63}.tpr
    |           |           |--------run3/
    |           |           |           |---------nes_{00..63}.tpr
    |           |           |--------run4/
    |           |                       |---------nes_{00..63}.tpr
    |           |--------stateB/
    |                       |--------run1/
    |                       |           |---------nes_{00..63}.tpr
    |                       |--------run2/
    |                       |           |---------nes_{00..63}.tpr
    |                       |--------run3/
    |                       |           |---------nes_{00..63}.tpr
    |                       |--------run4/
    |                                   |---------nes_{00..63}.tpr
    |------water/
                |--------stateA/
                |           |--------run1/
                |           |           |---------nes_{00..63}.tpr
                |           |--------run2/
                |           |           |---------nes_{00..63}.tpr
                |           |--------run3/
                |           |           |---------nes_{00..63}.tpr
                |           |--------run4/
                |                       |---------nes_{00..63}.tpr
                |--------stateB/
                            |--------run1/
                            |           |---------nes_{00..63}.tpr
                            |--------run2/
                            |           |---------nes_{00..63}.tpr
                            |--------run3/
                            |           |---------nes_{00..63}.tpr
                            |--------run4/
                                        |---------nes_{00..63}.tpr
```

The MUP1/ directory is organized in the same fashion as the c-Met/ directory.

Run each MD trajectory inside of the directory that it is located in. **Use the -deffnm keyword** for all of the MD runs. This will simplify the naming conventions of all of the output files. For instance, to run nes_14.tpr using 1 thread, you can use the command:
`gmx mdrun -nt 1 -deffnm nes_14`

That command will generate the following files which are needed for submission:
`nes_14.log`
`nes_14.xtc`
`nes_14.xvg`

PLEASE NOTE: For full credit, all 2048 simulations must be successfully completed. Failed simulations will be penalized in the scoring of this section.

Instructions - Calculating ΔΔG: Create a new python environment called 'pmx' in mamba. Install the following packages into this environment using the mamba install command:
`mamba install -c conda-forge numpy scipy matplotlib rdkit pandas git gromacs=2023.1 python=3.11`

Install the PMX software from this github repo: https://github.com/deGrootLab/pmx . Read the installation instructions carefully, and follow the procedure for installing PMX for Python 3.

Within this new "pmx" environment, you can run the provided python script for calculating the ΔΔG of binding, which accepts a directory as a variable.

To get the ΔΔG of binding between the two ligands in the MUP1 system, use the command:
`python calc_ddG.py MUP1/`

Likewise, to get the ΔΔG of binding between the two ligands in the c-Met system, use the command:
`python calc_ddG.py c-Met/`

The script `calc_ddG.py` will calculate the ΔΔG of binding based on two different methods (CGI and BAR). Perform these commands in the directory where the c-Met/ and MUP1/ directories reside. **Take the results and fill them into the provided spreadsheet.**

This script will also create files in their associated directories. **Include those files in your submission as well**. These files are `integA.dat`, `integB.dat`, `wplot.png`, and `results.txt`. They will appear in each section.

Requisite Files for Submission for Part 2:
part_2/
|------c-Met/

```
|       |------protein/
|       |           |--------integA.dat
|       |           |--------integB.dat
|       |           |--------wplot.png
|       |           |--------results.txt
|       |           |--------stateA/
|       |           |           |--------run1/
|       |           |           |           |-------nes_{00..63}.log
|       |           |           |           |-------nes_{00..63}.xvg
|       |           |           |           |-------nes_{00..63}.xtc
|       |           |           |--------run2/
|       |           |           |           |-------nes_{00..63}.log
|       |           |           |           |-------nes_{00..63}.xvg
|       |           |           |           |-------nes_{00..63}.xtc
|       |           |           |--------run3/
|       |           |           |           |-------nes_{00..63}.log
|       |           |           |           |-------nes_{00..63}.xvg
|       |           |           |           |-------nes_{00..63}.xtc
|       |           |           |--------run4/
|       |           |                       |-------nes_{00..63}.log
|       |           |                       |-------nes_{00..63}.xvg
|       |           |                       |-------nes_{00..63}.xtc
|       |           |--------stateB/
|       |                       |--------run1/
|       |                       |           |-------nes_{00..63}.log
|       |                       |           |-------nes_{00..63}.xvg
|       |                       |           |-------nes_{00..63}.xtc
|       |                       |--------run2/
|       |                       |           |-------nes_{00..63}.log
|       |                       |           |-------nes_{00..63}.xvg
|       |                       |           |-------nes_{00..63}.xtc
|       |                       |--------run3/
|       |                       |           |-------nes_{00..63}.log
|       |                       |           |-------nes_{00..63}.xvg
|       |                       |           |-------nes_{00..63}.xtc
|       |                       |--------run4/
|       |                                   |-------nes_{00..63}.log
|       |                                   |-------nes_{00..63}.xvg
|       |                                   |-------nes_{00..63}.xtc
|       |------water/
|                   |--------integA.dat
|                   |--------integB.dat
|                   |--------wplot.png
|                   |--------results.txt
```

```
|                   |--------stateA/
|                   |           |--------run1/
|                   |           |           |-------nes_{00..63}.log
|                   |           |           |-------nes_{00..63}.xvg
|                   |           |           |-------nes_{00..63}.xtc
|                   |           |--------run2/
|                   |           |           |-------nes_{00..63}.log
|                   |           |           |-------nes_{00..63}.xvg
|                   |           |           |-------nes_{00..63}.xtc
|                   |           |--------run3/
|                   |           |           |-------nes_{00..63}.log
|                   |           |           |-------nes_{00..63}.xvg
|                   |           |           |-------nes_{00..63}.xtc
|                   |           |--------run4/
|                   |                       |-------nes_{00..63}.log
|                   |                       |-------nes_{00..63}.xvg
|                   |                       |-------nes_{00..63}.xtc
|                   |--------stateB/
|                               |--------run1/
|                               |           |-------nes_{00..63}.log
|                               |           |-------nes_{00..63}.xvg
|                               |           |-------nes_{00..63}.xtc
|                               |--------run2/
|                               |           |-------nes_{00..63}.log
|                               |           |-------nes_{00..63}.xvg
|                               |           |-------nes_{00..63}.xtc
|                               |--------run3/
|                               |           |-------nes_{00..63}.log
|                               |           |-------nes_{00..63}.xvg
|                               |           |-------nes_{00..63}.xtc
|                               |--------run4/
|                                           |-------nes_{00..63}.log
|                                           |-------nes_{00..63}.xvg
|                                           |-------nes_{00..63}.xtc
|------MUP1/
       |------protein/
       |           |--------integA.dat
       |           |--------integB.dat
       |           |--------wplot.png
       |           |--------results.txt
       |           |--------stateA/
       |           |           |--------run1/
       |           |           |           |-------nes_{00..63}.log
       |           |           |           |-------nes_{00..63}.xvg
```

```
|           |           |           |-------nes_{00..63}.xtc
|           |           |--------run2/
|           |           |           |-------nes_{00..63}.log
|           |           |           |-------nes_{00..63}.xvg
|           |           |           |-------nes_{00..63}.xtc
|           |           |--------run3/
|           |           |           |-------nes_{00..63}.log
|           |           |           |-------nes_{00..63}.xvg
|           |           |           |-------nes_{00..63}.xtc
|           |           |--------run4/
|           |                       |-------nes_{00..63}.log
|           |                       |-------nes_{00..63}.xvg
|           |                       |-------nes_{00..63}.xtc
|           |--------stateB/
|                       |--------run1/
|                       |           |-------nes_{00..63}.log
|                       |           |-------nes_{00..63}.xvg
|                       |           |-------nes_{00..63}.xtc
|                       |--------run2/
|                       |           |-------nes_{00..63}.log
|                       |           |-------nes_{00..63}.xvg
|                       |           |-------nes_{00..63}.xtc
|                       |--------run3/
|                       |           |-------nes_{00..63}.log
|                       |           |-------nes_{00..63}.xvg
|                       |           |-------nes_{00..63}.xtc
|                       |--------run4/
|                                   |-------nes_{00..63}.log
|                                   |-------nes_{00..63}.xvg
|                                   |-------nes_{00..63}.xtc
|------water/
           |--------integA.dat
           |--------integB.dat
           |--------wplot.png
           |--------results.txt
           |--------stateA/
           |           |--------run1/
           |           |           |-------nes_{00..63}.log
           |           |           |-------nes_{00..63}.xvg
           |           |           |-------nes_{00..63}.xtc
           |           |--------run2/
           |           |           |-------nes_{00..63}.log
           |           |           |-------nes_{00..63}.xvg
           |           |           |-------nes_{00..63}.xtc
```

```
|           |--------run3/
|           |         |-------nes_{00..63}.log
|           |         |-------nes_{00..63}.xvg
|           |         |-------nes_{00..63}.xtc
|           |--------run4/
|                     |-------nes_{00..63}.log
|                     |-------nes_{00..63}.xvg
|                     |-------nes_{00..63}.xtc
|--------stateB/
           |--------run1/
           |         |-------nes_{00..63}.log
           |         |-------nes_{00..63}.xvg
           |         |-------nes_{00..63}.xtc
           |--------run2/
           |         |-------nes_{00..63}.log
           |         |-------nes_{00..63}.xvg
           |         |-------nes_{00..63}.xtc
           |--------run3/
           |         |-------nes_{00..63}.log
           |         |-------nes_{00..63}.xvg
           |         |-------nes_{00..63}.xtc
           |--------run4/
                     |-------nes_{00..63}.log
                     |-------nes_{00..63}.xvg
                     |-------nes_{00..63}.xtc
```