Running BE-META Simulations with GROMACS

(Written by Aidan Fike, updated for GROMACS 2018 by Jovan Damjanovic)

Before reading: This process is meant to be completed after you already have an equilibrated, solvated structure, represented by an equilibrated .gro file and corresponding topology file.

If you are not familiar with BE-META simulations as a concept, it will be good to first read “Escaping free-energy minima” by Alessandro Laio, followed by “A Bias-Exchange Approach to Protein Folding” by Stefano Piana and Alessandro Laio for a solid foundation. One of our group’s papers that may prove quite helpful is “Insights into How Cyclic Peptides Switch Conformations” by Sean M. McHugh *et al*.

**Part 1: Getting Access to PLUMED and Needed Modules**

PLUMED is a GROMACS-compatible plugin which includes an implementation of bias-exchange metadynamics. If your PLUMED is compiled in runtime mode, you will need to set your PLUMED\_KERNEL environment variable to point to the libplumedKernel.so file in the plumed/lib/ directory.

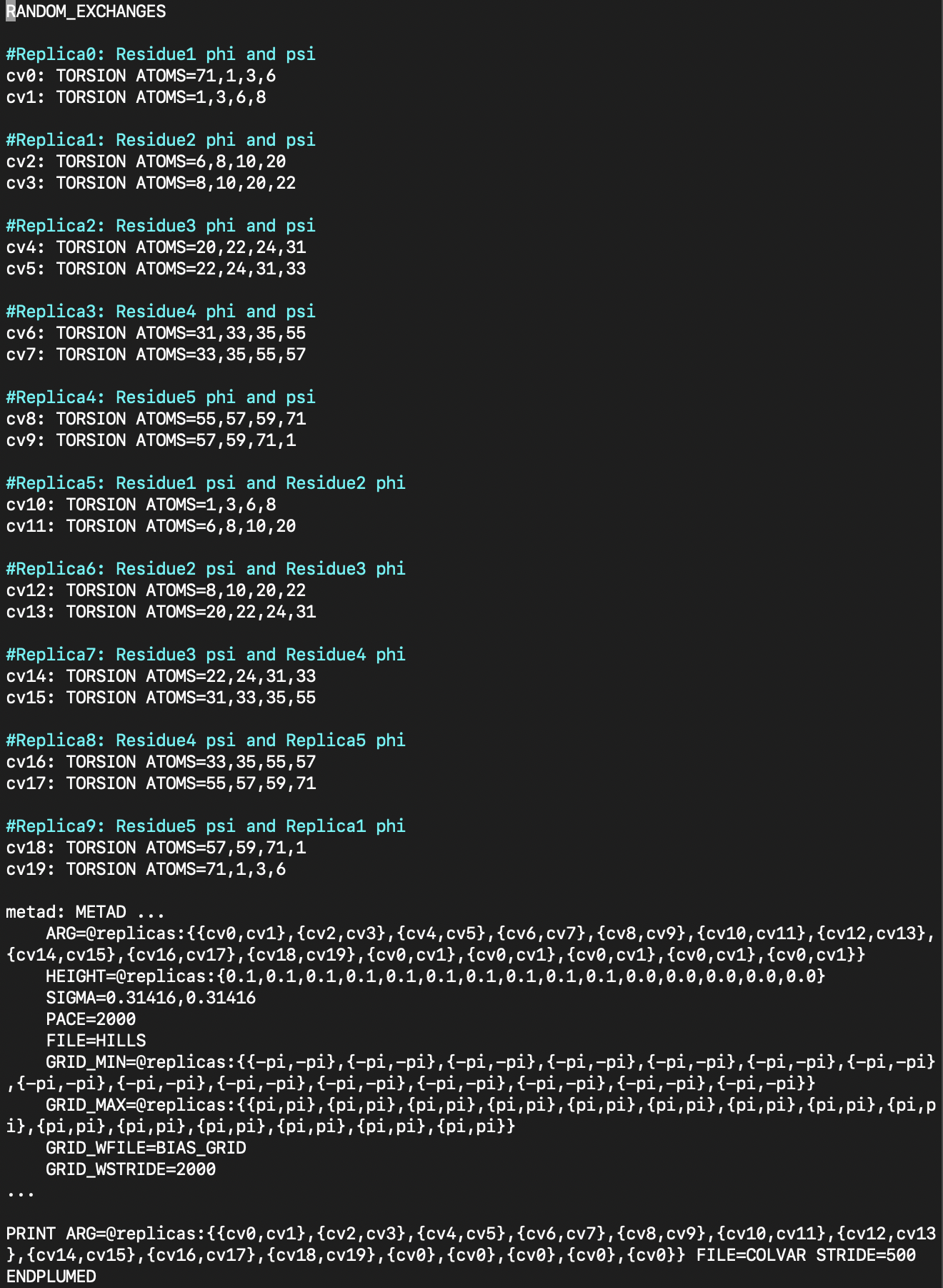
Make sure to also load the GROMACS build (previously patched with PLUMED) that you intend to use, along with all of its dependencies.

Additionally enter “echo $GMXLIB” to make sure that you are using your desired (and/or local) GMX library. Otherwise, set this environment variable accordingly.

Make sure you do not have multiple versions of OpenMPI or GROMACS loaded onto your path before running your simulation..

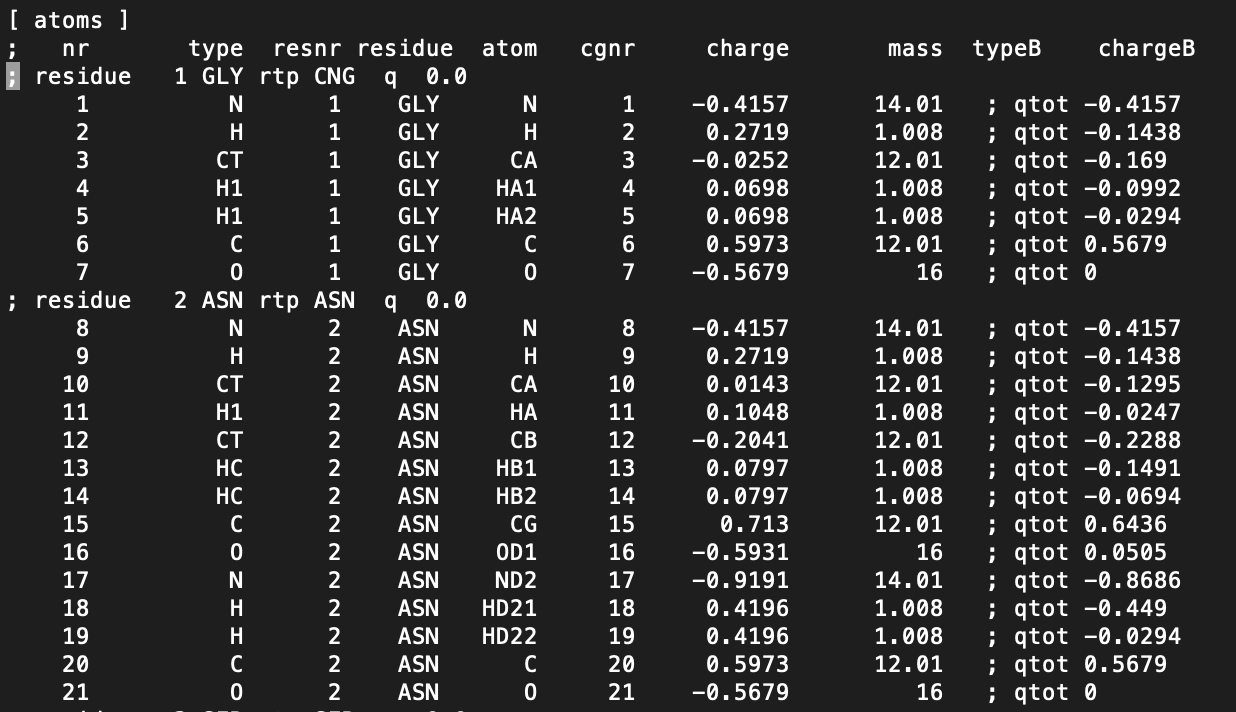
**Part 2: Setting up a BE-META simulation**

1. First, collect your now-equilibrated .gro and topology files that represent your solvated cyclic peptide
2. Next, create your .mdp file. These parameters should be nearly identical to the NPT file used for your equilibration. Make sure to only constrain h-bonds (note that this means bonds which contain a hydrogen, not hydrogen bonds). Additionally, adjust the time of the simulation to be 100 ns with a 2 fs timestep. For the sake of later analysis, it is necessary to output energy, .xtc, and .log file information every picosecond. An example .mdp file can be found in this repository and is titled prod\_npt\_bemeta.mdp.
3. In addition to the typical .gro, .top, and .mdp file combination that GROMACS utilizes, PLUMED requires a bemeta.dat file. This .dat file will contain information about the CVs (collective variables) that will be biased in each replica. Our group has found that one of the main obstacles to exploring the free energy landscape of a cyclic peptide is the ring strain introduced by cyclization. This makes it difficult for a given peptide to switch between conformations. To overcome this, the *φ*/*ψ* angles of the backbone are biased. Specifically, there will be replicas biasing the *φ*/*ψ* combination of each residue, as well as replicas biasing the *ψ* angle of each residue paired with the *φ* angle of the following residue. For a peptide like cGNSRV, you will have 10 biased collective variable pairs - 5 for *φi*/*ψi* combinations and another 5 for *ψi/φi*+1 combinations. To give this information to PLUMED, we use a bemeta.dat file. Here is an example of such a file:



The first line, RANDOM\_EXCHANGES, tells PLUMED to randomly swap trajectories between replicas. The timing of when these swaps happen is defined by the mdrun call, as will be seen later. The RANDOM\_EXCHANGES line tells PLUMED that when PLUMED does stop to exchange trajectories between replicas, it may do so between random, rather than sequential, pairs.

The next chunk of lines defines the collective variables used in the simulation. Cv2 is, for example, the *φ* angle of asparagine in GNSRV. This *φ* angle is defined by the collection of the 4 atomic indices that correspond to the relevant backbone atoms. We see that for the *φ* angle of asparagine, we are using indices {6,8,10,20}. Looking at our topology file below, we see that these indices correspond to {C N CA C}, the backbone atoms that constitute the *φ* angle.



At the bottom of the bemeta.dat file you will see the METAD and PRINT lines. This file, whose creation can also be somewhat automated using provided scripts, will be sent to a set of replicas in your BE-META simulation. The first 10 (or more generally, 2*N*) of these replicas will become the “biased replicas”. In each biased replica, a different pair of collective variables is being biased. In the METAD line of the bemeta.dat file, the “ARG=” indicates which two variables are being biased for each replica. The above example shows sets of CVs e.g. {cv0,cv1}, {cv2,cv3}, etc. The other parameters of METAD are the same across these biased replicas. SIGMA is the 2D width of the inserted Gaussian hills, and HEIGHT is the hill height. Notice that the last 5 HEIGHTs are 0. These 5 replicas are our neutral replicas, so there will be no Gaussian hills being inserted. Therefore, the canonical force field remains intact. To tell PLUMED that no Gaussian hills are being inserted, the hill height is zeroed out. PACE is how often these HILLS are inserted, in units of time steps. FILE is the file where the information about the inserted HILLS is deposited. Because there are 15 replicas in this case (10 biased and 5 neutral), we expect to see 15 different HILLS files being created. The PRINT command seen underneath METAD is simpler - it simply tells PLUMED to output the values of the indicated collective variables every STRIDE number of steps to FILE. We can reduce the computational time spent on evaluating the biases by storing the bias potentials on a grid defined by GRID\_MIN and GRID\_MAX, which are the min and max values your CVs can adopt. Since we are biasing dihedral angles, the values our CVs can adopt are from -pi to pi (-180° to 180°). More information about grid parameters can be found on the PLUMED site: <https://www.plumed.org/doc-master/user-doc/html/_m_e_t_a_d.html>

**Part 3: Running a BE-META simulation**

A .tpr file also needs to be created for each replica before running a BE-META simulation. You can grompp the .gro file from the last equilibration step (in our case, unrestrained NPT) to create one .tpr file and then copy and paste that .tpr file as many times as needed for each replica to have one. You may then run your simulation:

gmx\_mpi grompp -v -f prod.mdp -p GNSRV\_rsff2\_tip3p.top -c equil.gro -o start.tpr &> gmp.log

(copy and paste start.tpr to e.g., start0.tpr, start1.tpr, …,)

mpirun gmx\_mpi mdrun -v -plumed bemeta -multi 15 -replex 2500 -s start -deffnm prod

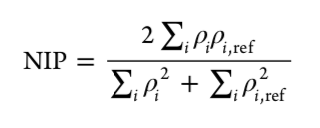
The -multi 15 argument tells PLUMED that you want 15 replicas. -replex 2500 refers to how many timesteps should be between replica exchanges. In this case, since we are working with 2 fs timesteps, we are performing exchanges every 5 ps.

**Part 4: Extending a Be-Meta simulation**

After running your first 100 ns simulation on both your S1 and S2 initial conformations, it is important to verify convergence. We do so using the normalized integrated product (NIP). You may want to refer to the analysis tutorial for more information here – obtain the principal components for your trajectory (starting with sines and cosines of the backbone dihedrals *φ*/*ψ*) and project the trajectory onto them; the top 3 should generally be enough.

In the resulting 3-dimensional space, you can easily compare the two trajectories. NIP = 1 would correspond to perfect convergence (identical ensembles) and NIP = 0 to no similarity at all.

To calculate NIP, first make a 50x50x50 grid of the 3-dimensional trajectory space for each of your s1 and s2 trajectories. Then, calculate the density of each of these cells: the fraction of frames it contains divided by the volume of the cell. Using these densities, you can then calculate NIP as



The “reference” density, ⍴ref, is simply the density of the other simulation’s corresponding cell. In other words, for S1, you are comparing to S2 (or rather, its last ~25-50 ns) as the “true” distribution, and for S2, you are comparing to S1 as the “true” distribution. You want to calculate NIP in simulation “chunks”, as the calculated structural ensemble is still being discovered at earlier times. That way, you can track the NIP evolution over time.

If the last 50 ns of your simulation have a low NIP (where “low” is usually less than 0.9), your system is not yet converged.

To extend your simulation, you only need to tweak a few elements of the simulation procedure explained above. Firstly, in your bemeta.dat files, you want to include the word “RESTART” on a line above RANDOM\_EXCHANGES. This will tell your simulation to read in the previous hill files as a starting point, and to append more information to the previous files, rather than begin new ones. You will also need to add onto your mdrun line the option

-cpi prod.cpt

This will take the checkpoint file provided by your last simulation, containing the position and velocities of your last frame, as the start of your new trajectory. It is good, however, to start each production run in a new directory (you may need to include -noappend in your mdrun statement to get GROMACS to start new output files).

**Part 5: Checking that your simulation completed successfully**

Here is a list of tips to check to see if your simulation completed successfully. This is by no means a comprehensive proof the simulation ran as intended, but is rather a quick sanity check to rule out major errors.

1. Check the output of mdrun (i.e., what GROMACS wrote to the console/a file) and make sure nothing crashed partway through.
2. Read through the log file of one of your neutral replicas and one of your biased replicas. You should also be reading the log file of your grompp process before your simulation begins. Check to see whether the computational performance is in line with your expectations
3. If you are extending your simulation, you should expect the size of your HILLS and COLVAR file to increase proportionally to the added time. Additionally, in the head of your biased log files, you should see a line under the section “PLUMED: Action METAD” that says that you are “Restarting from HILLS: \_\_\_\_ Gaussians Read”
4. Run dPCA/cluster analysis and NIP to see that your S1 and S2 simulations have converged to a similar solution that is biophysically logical.
5. Use VMD to watch through a trajectory of one of your neutral replicas. Make sure no bonds are broken, no *cis* bonds form, etc.
6. Read through your bemeta.dat file and check the TORSION ATOMS lines to see that the atoms are correctly chosen. Also, if you’re extending your simulation, make sure you do have a RESTART line at the top of the file.
7. Check your topology file to make sure that the RSFF2 script was properly applied to your system and that the once “missing” improper dihedrals on the peptide bond between your first and last residues were correctly added. Make sure the system has no net charge. This should optimally be done before your simulation begins.