Cyclic Peptide Solvating and Equilibration

**Part 3: Solvation and Equilibration**

By this point, you should have a working structure and topology file for your cyclic peptide. However, to properly simulate this structure, it is best to create an equilibrated box of explicit water surrounding it. To do so:

1. Run a vacuum energy minimization on your newly created protein structure. Use a steepest descent algorithm with 50,000 steps and no cutoff for vdw and coulomb. (Make sure to set pbc=no, or GROMACS will scream at you. Also make sure not to use PME electrostatics at this point if your system has a net charge, lest you get severe artifacts. Also (unless you’re using an older version of GROMACS, i.e., GROMACS 4.x.x) set cutoff-scheme=group in your .mdp file, or it will default to Verlet). Stop when maximum force is less than 10.0 kJ/mol/nm

gmx\_mpi grompp -v -f emin\_vac.mdp -c prot.gro -p cGNSRV\_rsff2\_tip3p.top

-o prot.tpr &> grompp.log

gmx\_mpi mdrun -v -s prot.tpr -deffnm vacmin &> mdrun.log

**Note**: An example emin\_vac.mdp file is included in the repository.

1. Next, add solvent around your protein. Specifically, first increase the size of your system such that there is 1.0nm between your peptide and the walls of the cubic boundary surrounding it. This can be done using

gmx\_mpi editconf -f vacmin.gro -o box.gro -bt cubic -d 1.0

gmx\_mpi solvate -cp box.gro -cs spc216.gro -p cGNSRV\_rsff2\_tip3p.top -o solv.gro

**Note**: If you need to use an older version of GROMACS (older than version 5.0), you will use the genbox command instead of the solvate command. This is because in updated versions of GROMACS, genbox command was replaced by solvate.

genbox -cp box.gro -cs spc216.gro -p cGNSRV\_amber99sb\_tip3p.top -o solv.gro

1. Then, (only) if your protein has a non-zero net charge, add ions to your system to neutralize it. If your protein is neutral, skip this step. In the case of GNSRV, since R has a +1 charge, we’ll want to add a chloride ion.

gmx\_mpi grompp -v -f emin\_vac.mdp -c solv.gro

-p cGNSRV\_rsff2\_tip3p.top -o solv.tpr

gmx\_mpi genion -s solv.tpr -p cGNSRV\_rsff2\_tip3p.top

-o solvion.gro -nname CL -nn 1

(**Note**: You can also just use the -neutral flag.)

In the cases where a positive ion is needed, -pname NA -np 1 is used to add a sodium ion instead. Additionally, if the peptide has, for example, a +2 net charge, you will want to use “-nn 2” to make sure you have 2 chloride ions.

1. Now, with our new RSFF2 force field, and the addition of our water molecules, we want to do another energy minimization of the entire solvated system. These parameters are roughly identical to those used in the production run except that energy minimization is on.

grompp -f emin\_pbc.mdp -c solvion.gro

-p cGNSRV\_rsff2\_tip3p.top -o start.tpr

mpirun mdrun -v -s start.tpr -deffnm emin

**Note**: An example emin\_pbc.mdp file is provided in the repository.

1. Now that we have an energy-minimized system, to get it ready for an actual production run, we want to slowly equilibrate the protein with the water surrounding it. This will be a gradual process. Begin with only simulating the water and hydrogens on your peptide. The rest of the “heavy” atoms on the peptide are position restrained. These position restraints are activated by using the -DPOSRES flag in the mdp file. This flag will trigger the inclusion of the posre.itp file that was created by pdb2gmx.

In this initial position restraint phase, run 50ps of NVT simulation, followed by 50ps of NPT simulation, each with 2fs timesteps. For temperature regulation, use a V-Rescale thermometer in two groups - one for Protein and the other for Non-Protein to prevent the hot solvent/cold solute problem. Each of these thermometers should have a reference temperature of 300K with a Tau of 0.1. Pressure coupling should be isotropic Parrinello-Rahman at 1 bar with Tau = 2.0 and compressibility of 4.5e-5 [1/bar]. PME should be used for electric interactions. For vdw: a 1.0nm cutoff with long range dispersion correction for energy and pressure. A LINCS algorithm should be used for constraints. For this equilibration phase, all bonds should be constrained on the protein and settle should be used for the water molecules. For the rest of the individual parameters, check out the mdp files in the repository.

One note is that the gen-vel parameter should be set to “yes” in the position restrained NVT equilibration, but “no” for the rest of the simulations you run. Otherwise you’ll lose key information from previous simulations. Also note that when you use the grompp command, you will need to use the -r flag along with your gro file.

1. Finally, once we finish this restrained equilibration, remove the position restraints, then run 100ps of NVT followed by 100ps of NPT. Other than removing the position restraints, the parameters of the mdp files should be identical to step 6, check out the example mdp files in the repository.

Now you should have an equilibrated, solvated system represented by a gro file (either solv.gro or solvion.gro) along with a topology file cGNSRV\_rsff2\_tip3p.top. Together, these can be simulated to find the behavior of your cyclic peptide (cGNSRV) in water. Congrats!

Before starting your production run, check the outputs of your equilibration steps. You need to load your trajectories into VMD and visualize them to make sure nothing strange has happened. Check that your water box looks normal (no cavities), and that each amino acid has the correct chirality and cis or trans peptide bond. Also check the location of any ions that were added to neutralize side chain charges.

After using manual inspection a few times for practice, you can use the scripts check\_trajComment.py and check\_chiralComment.py included in the repository to check your .gro files at the end of equilibration or solvation. For a treatment of the trajectory files (.xtc or .trr), see “4. Analysis tutorial”. When using these scripts, be sure to understand what use cases they are designed to handle as discussed in the README files, and what software dependencies they have; be sure to run them with **Python 3**. The scripts are designed to handle protein-only systems, so when providing a .gro file, remove any solvent water molecules (and accordingly update the value in the .gro file indicating the number of atoms). The specific atom positions are not important for these scripts, only the atom names and numbers.

Example usage:

python check\_trajComment.py –-gro equilProteinOnly.gro –-trj equilProteinOnly.gro –-cyclic True –-cutoff 120

python check\_chiralComment.py –-gro equilProteinOnly.gro –-trj equilProteinOnly.gro –-seq GNSRV

To start your full BE-META Simulation, read the document in this folder labeled “3. CP: Be-Meta Simulations”