**File Information:**

1. **Submit.bash**: this is the submitting a job to the cluster. This basically tells the cluster, hey when you’re ready and have the resources (all the #SBATCH conditions), run the .py script
2. **Run.py**: this is the running molecular dynamics (MD) self-assembly script. So if you look into the script it changes in and out of MD ensembles in various steps. It takes the start file (start.hoomdxml) as the topology and the path to the force field (you need two things to do a simulation: topology and force field)
3. **Start.hoomdxml**: this is the starting topology for running the simulation using hoomd
4. **Start.gsd**: this is the topology you can use vmd to look at since vmd doesn’t like .hoomdxml files on newer macs.
5. Directory **forcefield/cg**: this is where you will keep all the CG force field files
6. Directory **programs/hoomd-funcs**: this is where you keep some miscellaneous hoomd functions that are needed in the self-assembly script that hoomd didn’t have. This is pointed to in the submit.bash to include these functions.
7. Directory **simulation**: this is where all the experiments I do go. So within **simulation**, I would make a directory based upon what I’m simulating (based on this system: “mkdir **cernscholffa24**” . In the **cernscholffa24** directory (path: /users/cf2067/simulation/nscholffa24), I would put (scp) the submit.bash, the run.py, and the start files.
8. **Build-random-cg.py**: you don’t have to worry about this right now, but I included it for completeness. This script builds a CG randomized configuration where lipids are sandwiched between water slabs in the simulation box. When you run this script (typing “python build-random-cg.py” in the terminal window, the output is the two start files (start.hoomdxml and start.gsd). You don’t have to do anything with this right now as I have already sent you the start files you need to run a simulation. This script requires other files (a directory called **lipids** that has all the molecules and cg-FF local FF to build). Make sure within the build script you change the path to where you put the **lipids** directory. Note, in order to build a system, you have to create and run this script in an environment. I will include the lines at the end of this document you can type to create a conda environment called “building” where you can create systems.
9. Directory **lipids**: this is all the CG molecules you use when you run the build-random-cg.py script to make a randomized topology. You can look around (vim) the files if you want, but not super useful since I gave you start files.

**Things you need to do:**

Have two terminal windows open. One logged into DMOG and another in our One Drive directory called RunningOnDMOG (you can also have your own folder on your computer where you keep all your simulations for example I have: /Users/chloeframe/McCabeResearch/buildingfiles/cernscholffa24/trial1).

1. vmd start.gsd to see the starting structure (this is just a for funsies thing)
2. On DMOG in your home directory, make the three directories (**simulation**, **forcefield**, **programs**). “mkdir name” is the command you will use.
   1. Now locally, (see helpful commands below)
      1. scp the “cg” force field directory in the directory you made
      2. scp the “hoomd-funcs” to the programs directory you made
      3. scp the start files, run.py, and submit bash to the simulation study directory you made
3. Change all of the directory paths to connect properly with the right path for your account on DMOG (my account is cf2067).
   1. Two lines in run.py (FF path)

ff\_dir = '/users/cf2067/**forcefields**/cg' # all the .txt files are the non bonded parameters between that bead combination

bond\_pot, angle\_pot = load\_bonded\_coeffs(system, bond\_pot, angle\_pot, '/users/cf2067/**forcefields**/cg /cg-bond-coeffs\_stiff-tails-v3.p', units['D'], units['E'])

* 1. in submit.bash (email)

1. All the pieces should be there and connected so you can run a simulation by typing  
    “sbatch submit.bash”

**Useful terminal commands:**

* ls: list or show me what is in that directory
* mkdir foldername: make a directory
* cd path: to move around from folder to folder
* pwd: to figure out which directory you are in
* vim run.py: vim allows you to read into the file run.py; you can type “a” or “i” to start inserting text. To save and close out type “:wq”. To close out and not save changes type “:q!”.
* sbatch submit.bash: “sbatch” is the submit to the cluster command so “sbatch submit.bash” submits the whole process to the cluster to run
* squeue -u (your ID name): this will tell you if your job is running and for how long. For example: “squeue -u cf2067” will tell me which jobs are running
* this is locally, so not on DMOG, opening another terminal window: to push up a file (for example run.py) from your computer up to the cluster where you want to put it:   
  scp run.py [cf2067@dmog.hw.ac.uk:/users/cf2067/simulation/cernscholffa24](mailto:cf2067@dmog.hw.ac.uk:/users/cf2067/simulation/cernscholffa24)
* scp -r cg [cf2067@dmog.hw.ac.uk:/users/cf2067/forcefields](mailto:cf2067@dmog.hw.ac.uk:/users/cf2067/forcefields)
* scp -r hoomd-funcs [cf2067@dmog.hw.ac.uk:/users/cf2067/programs](mailto:cf2067@dmog.hw.ac.uk:/users/cf2067/programs)

\*\*scp -r (you need a -r anytime you move around a directory)

* To push a file up to the cluster: scp file username@cluster:/path
* To pull a file down from the cluster: scp [username@cluster:/path/file](mailto:username@cluster:/path/traj.dcd) locallocation
  + For example, pulling down a trajectory that is done running:   
    scp [username@cluster:/path/traj.dcd](mailto:username@cluster:/path/traj.dcd) /Users/chloeframe/McCabeResearch/buildingfiles/cernscholffa24/trial1/

Copy and paste below to create a local environment to be able to build a CG system with all the correct dependencies:

conda create --name building -c conda-forge -c mosdef -c omnia -c -janschulz python=3.7.5 numpy=1.17.3 mbuild=0.10.5 mdtraj=1.9.4 foyer=0.7.4 scipy=1.3.3 parmed=3.2.0 openmm=7.4.1 jupyter openbabel=3.0.0 gsd