Forcefield Creator

Version 2.1

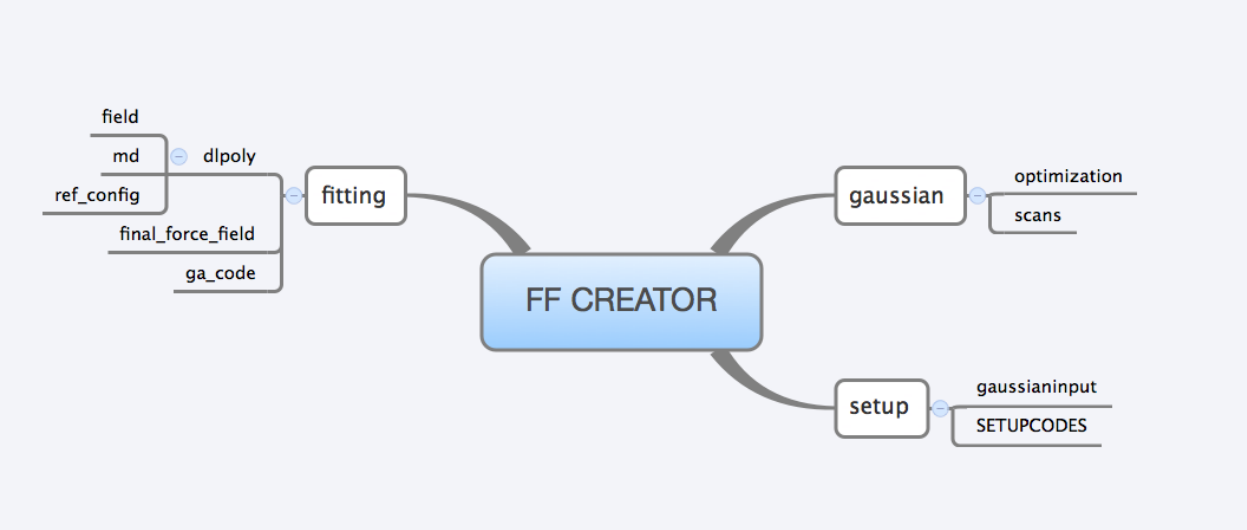
Z. L. Terranova

March 5, 2015

Contents

1 Overview

There are many ways to describe reality. Generating a set of interaction parameters for a metal organic framework (MOF) that can reasonable and accurately describe the underlying physics of the system is no simple task. These codes written in Fortran, Perl, and Bash scripts serve as a back-end wrapper to the Amber suite of programs to generate bond, angle, and dihedral information for MOFS. The generalized Amber force field (GAFF) was never designed for use with metals and



so force field parameters do not exist. To make these, we use a freely available genetic algorithm approach. The genetic algorithm selection scheme used is a tournament selection with a shuffling technique for choosing random pairs for mating. The routine includes binary coding for the individuals, jump mutation, creep mutation, and the option for single-point or uniform crossover.

Generating force field parameters is no easy business, and despite the fact that a certain set of parameters may give you the lowest sum of squares (SSQ), those parameters may not be physically applicable to your system. This is an arduous and frustrating procedure, however with the help of these scripts and codes it can hopefully be a little easier.

To install, run **build.sh** which will compile all the codes and put them in the correct spot.

2 Starting from a MOF-mimic structure

2.1 Initial setup

A MOF-mimic structure should contain a representative sample of the environment of the MOF. This part requires user input to determine what are the smallest, yet most physically relevant, pieces of information to include in this parameterization. Edit the *FFbuild.in* to include: the total number

of atoms, **natoms**, the name of the optimized pdb name, **pdbnam**, the unique atom that will be the center of the scan, **unatom**, the number of points to scan, **Npoints**, the total stepsize increment, **stepsize**, and the set folder your scans are held in, **scan**. The **aoi_pos** identifies the atom of interest AOI and the position it is in xyz coordinates. This is necessary to center that atom and wrap all other atoms around it.

For example, to scan a Cobalt MOF mimic, the initial structure would be called init.pdb and the unique atom name will be "co" at position 1. Assume we want to scan 5 steps in one dimension, and 5 steps in the opposite dimension, moving at 0.05 Angstroms. Setting Npoints=11 and stepsize=0.05 will scan the potential energy a total of 11 times: move 0.05 in the +x direction five times, move 0.05 in the -x direction five times, and calculate the energy on the 0.0 position = 11 total points. Note, Npoints must equal the number of points in one direction, a number of points in the opposite direction, +1 for the zero and will always be odd.

Execute an electronic structure of your system to obtain the charges. For the MOF systems in Gaussian 09, we use pop=nbo.

2.2 Setup the PES scans

After deriving the charges for your system, we now need to set up a potential energy scan. Begin by verifying the basis set (gaussianINPUT/basis) and ffHEADER file (gaussianINPUT/ffHEADER), the carverHEAD and carverTAIL (also located in gaussianINPUT/) are relevant to your system and file structure. The script **FFinit.sh** will use antechamber to create a .mol2 file, extract the charges, center the atom of interest, and most importantly generate the configurations necessary for the potential energy scan. After verifying all of the above information is accurate, execute **FFinit.sh**.

This will report where the folder containing all of the PES scans are located. Our group has computational time on carver.nersc.gov, and while the scripts are configured to be run on these machines, they can be altered for different systems. Securely copy (scp) this data to carver.nersc.gov and execute all of the job###.pbs files.

2.3 GAFFing the remaining atoms

While the PES scans are running, you can move on to setting up the rest of your system. As mentioned above, the script creates a .mol2 file. This file format contains some very helpful information in addition to the atomic coordinates such as gaff atom type, charges, and bonding information. Open this in your favorite program (Avogadro, GaussView, Gabedit, etc.) and you will notice certain bonds may be missing. Since the Amber suite was not designed for use with metals, it typically does not know how to deal with those bonds. It is necessary to draw in the remaining bonds and save the resulting .mol2 file, though depending on the program you use, make sure the gaff atom names (sixth column) are still present.

The gaff names were auto-generated via antechamber, and some may not be correct or not useful for your system. Take a look at gaff.dat for some inspiration and to see if any other gaff atom types are most appropriate for you and update the .mol2 file.

2.4 Generate the force_field.dat

Execute **gaffprep.sh**, which will create a force_field.dat. This file represents the base summary of all of the unique interactions present in your system with the AOI interactions being at the top of the categories. Ensure this file is complete with charges, masses, and all the interactions. Note, since we are fitting interactions involving the metal, it is ok to have them complete with 0.0000. Lastly, ensure that all of the LJ parameters supplied by hand are in units of σ. If you obtain values from gaff.dat, be sure to divide the R_{min} by $2^{1/6}$.

2.5 Make the DL_POLY FIELD file

Execute **createFIELD.sh** which will put everything in the correct spot and create a FIELD file necessary for DL_POLY. Check over the newly created FIELD file. This script will also create a ga.tmp file, which needs to be adjusted to the min and max of your choice.

3 After Gaussian PES scans are complete

3.1 Generate a target potential

Assuming all went well and you now have (Npoints x Npoints x Npoints) number of scans, it is time to generate a target potential. Run **createPES.sh**. This will go into the scan# folder as identified in the FFbuild.in, extract the energies, and put them into a target potential (target_pot). It is important to check over these values. Because the way the searching works, if the calculation did not converge the first time go into that log file and delete the first SCF Done occurrence. This script will also move the target_pot.dat in the correct place.

4 Ready to Run

4.1 Running the fit

You are ready to run if you have a target_pot and an appropriate FIELD file. To execute the fitting procedure, run **runFit.sh**. This script will take your ga.tmp file and put it in the proper location and execute the fitting procedure.

While it is running you can check on the progress by entering

```
grep SSQ ../fitting/best_fit.txt
```

5 Post Fit

Once complete, execute **createFinal.sh** which will grab the best_params, BESTPARAMS, a fig.png which is a snapshot of the first 10 potentials (and you can see more in Compare10.agr or pot.dat).

5.1 Further refinement

If you would like to run again or exert additional control or maximum and minimums, **createFinal.sh** copies the ga.inp it had used in the previous run back into the setup folder to ga.tmp. Additionally, any parameter changes need to be made in the new_params.dat and input files. Execute **reRun.sh**. This will build a new FIELD file from new_params.dat and rerun the fitting procedure according to the new ga.tmp file.

6 Closing Remarks

Though I have tried to generalize this procedure, it may fail in certain areas. I love a challenge and am more than willing to address any shortcomings in future versions. Please email me at zack.terranova@gmail.com