Build the topology of ion

Zhang Haomiao

HUST

December 19, 2016

1. PREPARE PDB
2. Generate PDB,Gaussian,GAMESS_US and fingerprint modeling
3. Perform the final modeling

PDB and mol2 file

AMBER NAMING scheme for residues HID HIE HIP $\rightarrow$ HIS
Prepare **mol2** file for non–standard residues

PDB ID: 1OKL

3 His residues and 1 MNS((5-(DimethyLamino)-1-Naphthalenesulfonamide)) ligand in its metal site

- Make sure no atoms use the same atom name in a certain residue, manually correct it before processing
- Make sure the residue name and atom name of the metal ion are all capitalized
- Make sure each metal ion or halide ion is treated separately as independent residue

Ligand file

1. Use ***reduce*** in AmberTools to add hydrogens to ligands. (GaussainView for manually).

```
reduce MNS.pdb > MNS_H.pdb
```

2. Delete the hydrogens which connect to the ligating nitrogen atom of zinc ion. » `MNS_fixed_H.pdb`
3. Use ***antechamber*** to generate mol2 file for non_standard residues. (`AM1-BCC` to generate the charges, ligand has a charge -1 , `GAFF` atom types)

```
antechamber -fi pdb -fo mol2 -i MNS_fixed_H.pdb -o MNS_pre.mol2 -c bcc -pf y -nc -1
```

Note there is no "du" atom type in MNS_pre.mol2 file , rename it to `MNS.mol2`

4. Perform the following command to obtain frcmod file for ligand

```
parmchk2 -i MNS.mol2 -o MNS.frcmod -f mol2
```

Please make sure there is one and only one blank line after each parameter section » `MNS.frcmod`

Metal ion and water

1. Copy Zn ion into one single PDB file (`ZN.pdb`)
2. Use ***antechamber*** to generate mol2 file

```
antechamber -fi pdb -fo mol2 -i ZN.pdb -o ZN_pre.mol2 -at amber -pf y
```

3. Change the atom type and charge in the `ZN_pre.mol2` file, treat Zinc ion with atom "ZN" and charge 2.0 » `ZN.mol2`
4. Keep crystal water during modeling, copy it into `WAT.pdb`
5. Use ***tleap*** to add hydrogen atoms to water

```
tleap -s -f wat_tleap.in > wat_tleap.out
```

6. use ***antechamber*** to generate mol2 files for water molecules.Using AMBER atom types

```
antechamber -fi pdb -fo mol2 -i WAT_H.pdb -o WAT.mol2 -at amber -c bcc -pf y
```

» `WAT_H.pdb WAT.mol2`

Combine PDB files into PDB file

1. use the webserver H++ to add hydrogen atoms to the PDB file, H++ will delete the non-standard residues during the modeling process, this PDB file will use an AMBER naming scheme for the residues.

```
ambpdb -p 0.15_80_10_pH6.5_1OKL.top -c 0.15_80_10_pH6.5_1OKL.crd > 1OKL_Hpp.pdb
```

2. Palce the standard residues, metal ion , ligand and water
3. use cat to combine the pdb file

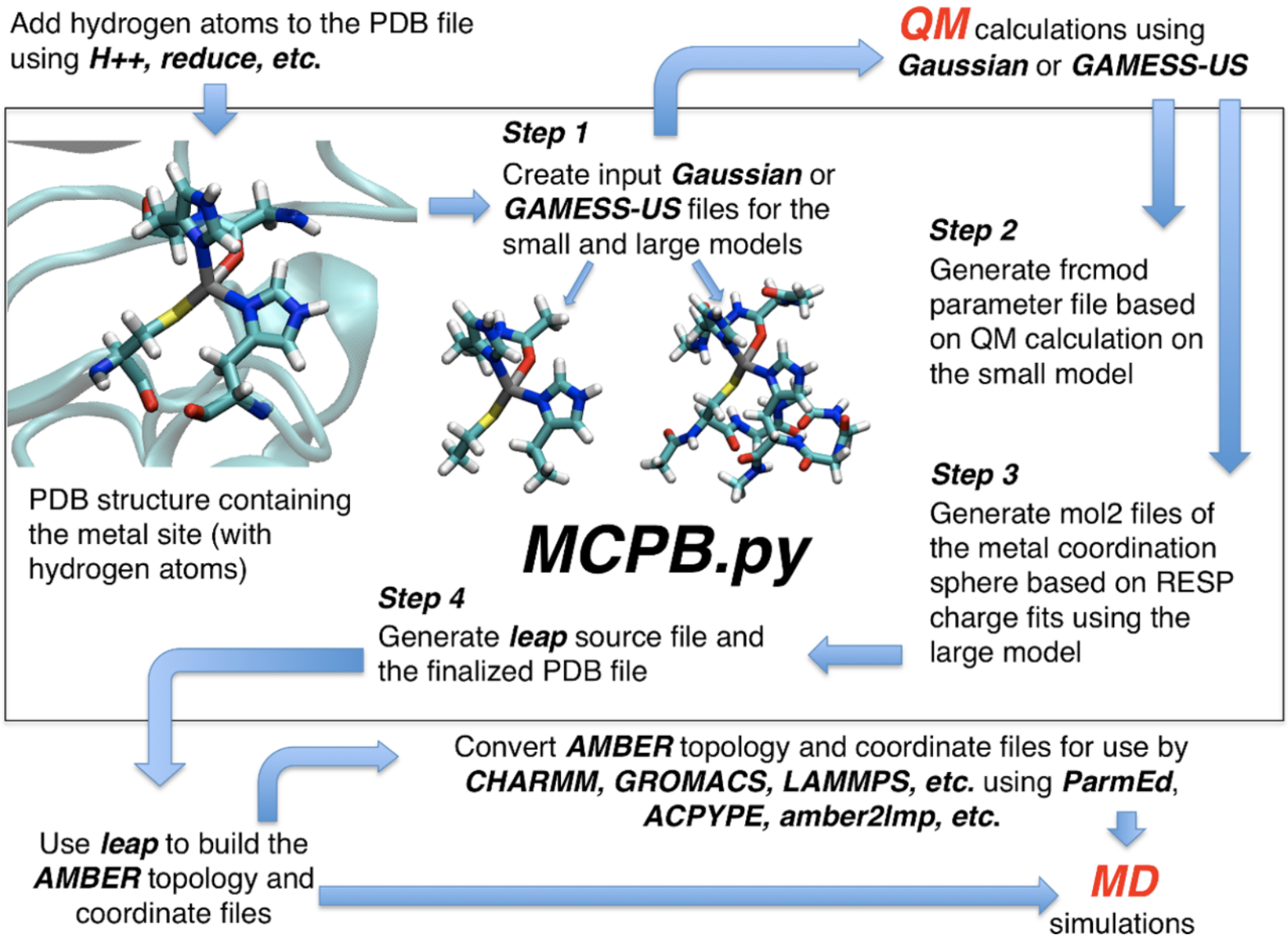
```
cat 1OKL_Hpp_fixed.pdb ZN.pdb MNS_H_fixed.pdb > 1OKL_H.pdb
```

4. using **pdb4amber** to renumber PDB file

```
pdb4amber -i 1OKL_H.pdb -o 1OKL_fixed_H.pdb
```

» PDB file : 1OKL_fixed_H.pdb
» mol2 file MNS.mol2 ZN.mol2

Workflow of MCPB.py



Generate file

1. INPUT FILE : 1OKL.in , using Gaussian09 , add "software_version g09" as a line into the input file
   - large_opt equal 1 to optimize the hydrogen postions of the large model
   - ```
     MCPB.py -i 1OKL.in -s 1
     ```
   - OUTPUT PDB,fingerprint files of small, standard and large models
   - OUTPUT Gaussian input file of small and large models
2. Run Gaussian09, perform geometry optimization, force constant calculation

   ```
   g09 < 1OKL_small_opt.com > 1OKL_small_opt.log
   g09 < 1OKL_small_fc.com > 1OKL_small_fc.log
   ```
3. generate fchk file for small model

   ```
   formchk 1OKL_small_opt.chk > 1OKL_small_opt.fchk
   ```
4. Perform the Merz-Kollman RESP charge calculation for the large model

   ```
   g09 < 1OKL_large_mk.com > 1OKL_large_mk.log
   ```
   ```

- **Seminario method** to generate force field parameters

```
MCPB.py -i 1OKL.in -s 2
```

» 1OKL_mcpbpy.frcmod

- use the ChgModB to perform the RESP charge fitting and generate the mol2 files for the metal site residues

```
MCPB.py -i 1OKL.in -s 3
```

» HD1.mol2, HD2.mol2, HE1.mol2, ZN1.mol2 MS1.mol2

- Using mol2 file in leap modeling,generate tleap input file

```
MCPB.py -i 1OKL.in -s 4
```

» 1OKL_mcpbpy.pdb, 1OKL_tleap.in

- use tleap to generate the topology and coordinate files

```
tleap -s -f 1OKL_tleap.in > 1OKL_tleap.out
```

» 1OKL_solv.prmtop, 1OKL_solv.inpcrd

Check the modeling

- use VMD to do a check about whether the coordiante bonds to metal ion exist in the topology file select **Graphics→Representations**,type same **residue as within 3 of resname ZN1**,if all the four coordination bonds to zinc ion, then the outcome is correct.
- use cpptraj to do a check about the atom numbering issue

```
cpptraj -p 1OKL_solv.prmtop
```

- Finally,use ParmEd to check the metal site parameters

```
parmed -i mcpbpy_parmed.in -p 1OKL_solv.prmtop
```

A) The bond force constants between an metal ion and its ligating atoms are less than 200 kcal/(mol*Angstrom$\hat{2}$), and the eqlibirum bond distances are less than 2.8 Angstrom;
B) The angle force constants related to the metal ion are usually less than 100 kcal/(mol*Rad$\hat{2}$) while the equlibirum angle values are bigger than 100 Degree;
C) All or most of the dihedral potential barriers are zero for metal involved dihedrals;

Check the modeling

- use VMD to do a check about whether the coordiante bonds to metal ion exist in the topology file select **Graphics→Representations**,type same **residue as within 3 of resname ZN1**,if all the four coordination bonds to zinc ion, then the outcome is correct.
- use cpptraj to do a check about the atom numbering issue

```
cpptraj -p 1OKL_solv.prmtop
```

- Finally,use ParmEd to check the metal site parameters

```
parmed -i mcpbpy_parmed.in -p 1OKL_solv.prmtop
```

D) The RESP charge of the metal ion are less than its oxidation state, usually even less than +1;
E) The LJ Radius of one metal ion is usually bigger than 1.0 Angstrom.