Use of the General Amber Force Field GAFF or GAFF2, and the AmberFF14SB or AmberFF19SB force fields

Tutorial by:
F. Wang,(1) J.-P. Becker, (1) P. Cieplak,(2) and F.-Y. Dupradeau(1)

(1) Université de Picardie - Jules Verne, Amiens
(2) Sanford-Burnham Institute for Medical Research, La Jolla, CA

November 2014
Updated August 2022

The quantum mechanics (QM) Gaussian, GAMESS or Firefly program is selected during the input submission procedure using the web interface: one can use the public account or a private account: See http://q4md-forcefieldtools.org/REDServer-Development/faq.php#1

Input molecules of PyRED are provided in the PDB file format: See http://q4md-forcefieldtools.org/REDServer-Development/Documentation/readme.txt

-I- APPROACH COMPATIBLE WITH GAFF OR GAFF2

A single PyRED job approach is presented to generate a force field compatible with GAFF or GAFF2

-II- APPROACH COMPATIBLE WITH AMBER14FFSB OR AMBER19FFSB

A more detailed approach involving two PyRED jobs is presented to generate a force field compatible with AMBER14FFSB, AMBER19FFSB or whatever Amber force field versions...

-I- APPROACH COMPATIBLE WITH GAFF or GAFF2 [1]

-I.1- Define specific keywords in the System.config file:

See http://q4md-forcefieldtools.org/REDServer-Development/Documentation/System.config

```
# As not provided use the default values for OPT_Calc, MEPCHR_Calc and Freq_Calc
```
Select the charge model and the force field:
```
# RESP-A1 is the default; so this keyword is not needed...
CHR_TYP     = RESP-A1
```
You may decide to select the RESP-X1 charge model instead of RESP-A1
AMBERFF10 is the default; so this keyword is not needed:
```
FFPARM     = AMBERFF10
```
Finally, you may decide to adapt the METHOD_OPTCALC, BASSET_OPTCALC,
METHOD_MEPCALC and BASSET_MEPCALC keywords for GAFF2
(these keywords are limited to the use of the Gaussian program)

-I.2- Define specific keywords in the Project.config file:

See http://q4md-forcefieldtools.org/REDServer-Development/Documentation/Project.config

The user has to define the total charge and spin multiplicity of the molecules involved in force field generation if they are different from zero and one, respectively. The user can also define bond(s), angle(s) and/or dihedral(s), which can be constrained during geometry optimization. Charge constraints can be requested during the charge fitting step to lead to molecular fragments as well, etc…

```
# An informative title for methanol:
MOLECULE1-TITLE        = Methanol
# Total charge = 0 and spin multiplicity = 1 are the default values used by PyRED
#    So these two keywords are not needed...
MOLECULE1-TOTCHARGE    = 0
MOLECULE1-SPINMULT     = 1
```
Then, the user has to provide the atom types for the molecules provided as input files. In this case, the dictionary of atom types developed in PyRED is not used, and the GAFF atom types (lower case) provided in the `MOLECULE1-ATMTYPE` keyword are read:

For instance for methanol:
The atom type order is that in the Mol_red1.pdb input file:
This overwrites the default 'FFPARM' keyword value in the System.config file
```
MOLECULE1-ATMTYPE      = c3 h1 h1 h1 oh ho
```

-I.3- Use the gaff.dat or gaff2.dat file as a frcmod.user file:

The gaff.dat or gaff2.dat force field parameter file has to be provided as a *frcmod.user* file:
```
cp $AMBERHOME/dat/leap/parm/gaff.dat  frcmod.user     or
cp $AMBERHOME/dat/leap/parm/gaff2.dat frcmod.user
```

-I.4- Create the archive file to upload the different PyRED inputs

See http://q4md-forcefieldtools.org/REDServer-Development/Documentation/Create-archive.php

```
tar -zcvf Archive.tgz Mol_red*.pdb Project.config System.config frcmod.user
```
→ Submit to https://upjv.q4md-forcefieldtools.org/REDServer-Development/

---

[1] GAFF: https://doi.org/10.1002/jcc.20035; Wang et al. *J. Comput. Chem.* **2004**, 25, 1157.
    GAFF2: Wang, Last version of the AmberTools...

-II- APPROACH COMPATIBLE WITH AMBERFF14SB or AMBERFF19SB [2]

Let's take the example of generating the central fragment of ζ-N-methyl-L-Lysine in this tutorial:
ζ-N-methyl-L-Lysine: see https://en.wikipedia.org/wiki/Methyllysine
New central fragment: see http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#17
Let's start from the MeCO-ζ-N-methyl-*L*-Lysine-NHMe dipeptide; *i. e.* the backbone of ζ-N-methyl-L-Lysine is capped by the acetyl (MeCO, ACE) and NH-methyl (NME) groups: two 'trans' peptide bonds are formed (dihedral angle ω ~ 180°). Choose also the conformation(s) with characterized φ, ψ, and χ1-χ4 dihedral angle values to be involved in charge derivation (not discussed in this tutorial).

Create the `Mol_red1.pdb` PDB input file for the ζ-N-methyl-L-Lysine dipeptide: hydrogen atoms are generally grouped at the end of PDB file because they are added after all the heavy atoms, when constructing the PDB file. Setting '`MOLECULE1-ATMREORDR = ON`' allows modifying the atom order so that the hydrogen atoms are located after the heavy atoms they are bound to in the atom order. Such order is convenient to define the atom indexes involved in charge constraints used to generate molecular fragments, and to better understand the RESP input/output files. However, it means also that the atom order in the input and output PyRED files are different. To correct these input/output differences, the force field for the central fragment of the dipeptide of ζ-N-methyl-*L*-Lysine is carried out in two successive PyRED jobs:

-II.1- Define specific keywords in the Project.config and System.config files for the first PyRED job to reorder the atoms and to check the stationary point:

Pay attention to the atom and residue names in the `Mol_red1.pdb` input file: if the central fragment of the ζ-N-methyl-L-Lysine dipeptide is present in a macromolecule downloaded from the protein data bank and loaded in the LEaP program from the AmberTools with the chosen force field, the atom and residue names in the `Mol_red1.pdb` input; *i. e.* these in the `Mol_m1/Mol-sm_m1-c1.mol2` force field library have to match these in the PDB macromolecule file; please read:
http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#3

The System.config file:
```
OPT_Calc            = ON
MEPCHR_Calc         = ON
# Checks the nature of the stationary point found after geometry optimization
#   (time consuming step; search for unwanted negative frequencies:
#    select another conformation if negative frequency/ies are found)
Freq_Calc           = ON
CHR_TYP             = RESP-A1
FFPARM              = AMBERFF10
```
The Project.config file:
```
MOLECULE1-TITLE       = 6N-methyl-L-Lysine_dipeptide
MOLECULE1-TOTCHARGE   = 1
MOLECULE1-SPINMULT    = 1
MOLECULE1-ATMREORDR   = ON
```

→ Create the archive file for the PyRED inputs, run the first PyRED job at https://upjv.q4md-forcefieldtools.org/REDServer-Development/, download the '`Pxxxx.tar.bz2`' archive file and open this new archive file.

-II.2- Define specific keywords in the 'config' files for the second PyRED job to generate the central fragment for the input dipeptide in agreement with AMBERFF14SB or AMBERFF19SB:

Select pieces of information from the first PyRED job, and prepare the second job in a new directory:
```
mkdir Job2
```
The geometry of Mol_red1-out.ent is already optimized:
```
cp Pxxxx/Data-R.E.D.Server/Data-Default-Proj/Mol_red1-out.ent  Job2/Mol_red1.pdb
cp Pxxxx/Data-R.E.D.Server/Mol_m1/JOB1-gau_m1-1.out          Job2/Mol_red1.log
```

The geometry optimization step was done in the first job:
```
echo 'OPT_Calc = OFF'   >  Job2/System.config
echo 'CHR_TYP = RESP-A1  >> Job2/System.config
echo 'FFPARM = AMBERFF10 >> Job2/System.config
```

Selected important keywords from the first job in agreement with the new atom order:
```
egrep  'MOLECULE1-TITLE|MOLECULE1-TOTCHARGE|MOLECULE1-ATMTYPE' Pxxxx/Data-R.E.D.Server/Data-
Default-Proj/Project-out.config  > Job2/Project.config
```

Let's have a look at the AMBER10 atom types generated in the first PyRED job:
```
MOLECULE1-ATMTYPE = CT HC HC HC C O N H CX H1 CT HC HC CT HC HC CT HC HC CT HP HP N3 H H CT
HP HP HP C O N H CT H1 H1 H1
```
One finds the 'CX' atom types for the Cα carbon atom defined in Amber10; one also find the 'CT' types for the Cβ, Cγ, Cδ and Cε carbon atoms, but one wants the new 'C8' one instead defined in AMBERFF14SB. Let's rename the carbon atom types for the residue side chain: with the new atom order the four methylene groups are easily identified:
```
sed -i 's|CT HC HC CT HC HC CT HC HC CT HP HP|C8 HC HC C8 HC HC C8 HC HC C8 HP HP|'
Job2/Project.config
```

Add the two intra-molecular charge constraints with the remove 'flag' to create the central fragment; the atom indexes are easily found when atoms are reordered (the six first indexes for the first charge constraint, and the last six indexes for the second constraint):
```
echo 'MOLECULE1-INTRA-MCC1 = 0.0 |  1  2  3 4  5   6 | Remove' >> Job2/Project.config
echo 'MOLECULE1-INTRA-MCC1 = 0.0 | 32 33 34 35 36 37 | Remove' >> Job2/Project.config
```

Copy the force field parameters from the AmberTools to match the force field one wishes to match:
```
cp $AMBERHOME/dat/leap/parm/frcmod.ff14SB Job2/frcmod.user
```

Create the archive file for the second job and submit to REDServer-Development:
```
tar   -zcvf   archive.tgz   Job2/Mol_red1.pdb   Job2/Mol_red1.log   Job2/System.config
Job2/Project.config Job2/frcmod.user
```

Remarks:
- One can adopt a similar strategy for all the present/future adaptations of the Amber10 force field. [2]
- To generate the N-terminal and and C-terminal fragments besides the central one, see also:
   http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#26

[2] https://doi.org/10.1016/j.bpj.2009.04.063; Wickstrom et al. *Biophysical J.* **2009**, 97, 853.
   https://doi.org/10.1002/prot.22711; Lindorff-Larsen et al. *Proteins* **2010**, 78, 1950.
   https://doi.org/10.1021/acs.jctc.5b00255; Maier et al. *J. Chem. Theory Comput.* **2015**, 11, 8, 3696.
   https://doi.org/10.1021/acs.jctc.9b00591; Tian et al. *J. Chem. Theory Comput.* **2020**, 16, 1, 528.

Last update of this tutorial: August 20th, 2022.