Introduction

The AMBER ff14ipq force field includes a new charge derivation protocol that is laborious but straightforward to carry out. This tutorial explains how to derive atomic charges and dihedral parameters for the nonstandard amino acid norleucine, including its nonterminal, N-terminal, and C-terminal forms.

1.0 Generating Initial Parameters

The IPolQ method of deriving atomic charges is an iterative optimization protocol that requires an initial set of parameters. These parameters include the structure, connectivity, atom types, and charges of the amino acid which may be provided to AMBER in the form of a `lib` file. Depending on the atom types present in the molecule, an `frcmod` file containing additional parameters not present in the ff14ipq force field may be required as well. If these are already available for the amino acid of interest (e.g. parameters previously derived for use with the ff9X/ff1X family of force fields), Section 1.1 may be skipped.

1.1 Preparing Initial mol2 Files

The first step is to to obtain or construct initial structures of the nonterminal, N-terminal, and C-terminal forms of the amino acid. The structures may be stored most conveniently in the mol2 format, which can store the atom type and charge information required by AMBER. It is helpful to follow atom naming conventions consistent with AMBER force fields, these may be read from the ff14ipq residue templates at `$AMBERHOME/dat/leap/lib/amino14ipq.lib`. A sample `mol2` file of norleucine (NLE) using these conventions follows:

```
@<TRIPOS>MOLECULE
NLE
  19  18   1   0
SMALL
NO_CHARGES

@<TRIPOS>ATOM
   1 N     -1.9680   0.0710   0.3120 N    1 NLE    0.000000
   2 H     -1.4810  -0.3330   1.1060 H    1 NLE    0.000000
   3 CA    -1.2060   1.2220  -0.1620 CA   1 NLE    0.000000
   4 HA    -1.3700   1.3210  -1.2370 HA   1 NLE    0.000000
   5 CB    -1.7010   2.5050   0.5400 CB   1 NLE    0.000000
   6 HB2   -2.7830   2.4620   0.6610 HB2  1 NLE    0.000000
   7 HB3   -1.2860   2.5590   1.5490 HB3  1 NLE    0.000000
   8 CG    -1.3600   3.7940  -0.2280 CG   1 NLE    0.000000
   9 HG2   -1.8470   3.7630  -1.2030 HG2  1 NLE    0.000000
  10 HG3   -0.2880   3.8500  -0.4220 HG3  1 NLE    0.000000
  11 CD    -1.8050   5.0560   0.5250 CD   1 NLE    0.000000
  12 HD2   -2.8560   4.9750   0.8000 HD2  1 NLE    0.000000
  13 HD3   -1.2490   5.1380   1.4590 HD3  1 NLE    0.000000
  14 CE    -1.6020   6.3320  -0.2980 CE   1 NLE    0.000000
  15 HE1   -1.8950   7.2150   0.2720 HE1  1 NLE    0.000000
  16 HE2   -0.5590   6.4580  -0.5890 HE2  1 NLE    0.000000
  17 HE3   -2.2040   6.3150  -1.2090 HE3  1 NLE    0.000000
  18 C      0.2920   1.0090   0.0910 C    1 NLE    0.000000
  19 O      0.7170   0.4370   1.0960 O    1 NLE    0.000000
@<TRIPOS>BOND
   1   1   2   1
   2   1   3   1
   3   3   4   1
   4   3   5   1
```

```
   5   3  18   1
   6   5   6   1
   7   5   7   1
   8   5   8   1
   9   8   9   1
  10   8  10   1
  11   8  11   1
  12  11  12   1
  13  11  13   1
  14  11  14   1
  15  14  15   1
  16  14  16   1
  17  14  17   1
  18  18  19   1
@<TRIPOS>SUBSTRUCTURE
   1 NLE  1 TEMP 0 **** **** 0 ROOT
```

Note that this does not include atom type or charge information. A straightforward method of obtaining initial values for these is via the framework already in place for the AMBER ff9X/ff1X family of force fields. The necessary adjustments to be made to these parameters prior to optimization are fairly minimal. Charges and atom types may be obtained using AmberTools' `antechamber` program:

```
antechamber -fi mol2 \              # Input file format
            -i  norleucine.mol2 \   # Input file name
            -rn NLE \               # Name of residue; amino acids in AMBER use
                                    #   three-letter codes; prepended with N or C
                                    #   for the N- and C- terminal versions
            -nc 0 \                 # Set net charge of molecule to 0; for charged
                                    #   residues settings of +1 or -1 are
                                    #   appropriate
            -c  bcc \               # Calculate charges using the AM1/BCC
                                    #   semi-empirical method; while for final
                                    #   parameters to be used with AMBER ff9X/1X
                                    #   force fields HF/6-31G* is preferred, for
                                    #   the purpose of selecting an initial set
                                    #   of charges AM1/BCC is sufficent
            -eq 1 \                 # Use equal charges for equivalent atoms
                                    #   based on connectivity
            -at amber \             # Use atom types consistent with the AMBER
                                    #   ff9X/ff1X family or force fields, rather
                                    #   than the General AMBER Force Field
                                    #   (GAFF), this is appropriate for
                                    #   nonstandard amino acids
            -j  5 \                 # Use atom and part bond type prediction
            -s  2 \                 # Be verbose
            -pf y \                 # Remove intermediate files
            -fo mol2 \              # Output file format
            -o  norleucine_bcc.mol2 # Output file name
```

This produces an output `mol2` file including the atom names and partial charges. The updated portion of the file is reproduced below:

```
@<TRIPOS>ATOM
   1 N     -1.9680   0.0710   0.3120 DU   1 NLE   -0.616000
```

```
   2 H     -1.4810  -0.3330   1.1060 H    1 NLE    0.341800
   3 CA    -1.2060   1.2220  -0.1620 CT   1 NLE    0.110600
   4 HA    -1.3700   1.3210  -1.2370 H1   1 NLE    0.112700
   5 CB    -1.7010   2.5050   0.5400 CT   1 NLE   -0.110400
   6 HB2   -2.7830   2.4620   0.6610 HC   1 NLE    0.055200
   7 HB3   -1.2860   2.5590   1.5490 HC   1 NLE    0.055200
   8 CG    -1.3600   3.7940  -0.2280 CT   1 NLE   -0.074400
   9 HG2   -1.8470   3.7630  -1.2030 HC   1 NLE    0.045200
  10 HG3   -0.2880   3.8500  -0.4220 HC   1 NLE    0.045200
  11 CD    -1.8050   5.0560   0.5250 CT   1 NLE   -0.080400
  12 HD2   -2.8560   4.9750   0.8000 HC   1 NLE    0.039700
  13 HD3   -1.2490   5.1380   1.4590 HC   1 NLE    0.039700
  14 CE    -1.6020   6.3320  -0.2980 CT   1 NLE   -0.093100
  15 HE1   -1.8950   7.2150   0.2720 HC   1 NLE    0.034700
  16 HE2   -0.5590   6.4580  -0.5890 HC   1 NLE    0.034700
  17 HE3   -2.2040   6.3150  -1.2090 HC   1 NLE    0.034700
  18 C      0.2920   1.0090   0.0910 CZ   1 NLE    0.471800
  19 O      0.7170   0.4370   1.0960 O    1 NLE   -0.446900
```

For the N- and C- terminal forms of norleucine, it is more straightforward to set the atom types and charges manually, as discussed in the next section.

1.2 Adjusting mol2 Files for Consistency with ff14ipq

Several modifications are necessary to prepare the inital parameters for IPolQ charge derivation.

First, the atom types must be adjusted to be consistent with ff14ipq. The backbone atoms of ff14ipq use the new types CX and OD for CA and O. For norleucine, we also change the types of the internal side chain carbons from CT to 2C, the type used for side-chain carbons connected to two other carbons. The terminal carbon is left as CT, consistent with other methyl carbons in AMBER force fields.

Second, the backbone charges must be adjusted. ff14ipq uses single sets of charges for the N, H, C, and O atoms of the backbone for neutral, positively charged, and negatively charged amino acids. The charges of these atoms of norleucine may be set to these shared values and fixed during the charge-fitting process. After making these adjustments, the net charge of the molecule is no longer 0, and we must apply the necessary balance of charge to non-backbone atoms. For our purpose of generating an initial set of charges; we may simply divide the residual charge equally between the remaining atoms The updated portion of the `mol2` file follows:

```
@<TRIPOS>ATOM
   1 N     -1.9680   0.0710   0.3120 N    1 NLE   -0.49998
   2 H     -1.4810  -0.3330   1.1060 H    1 NLE    0.31825
   3 CA    -1.2060   1.2220  -0.1620 CX   1 NLE   -0.053189
   4 HA    -1.3700   1.3210  -1.2370 H1   1 NLE    0.142811
   5 CB    -1.7010   2.5050   0.5400 2C   1 NLE   -0.189189
   6 HB2   -2.7830   2.4620   0.6610 HC   1 NLE    0.096311
   7 HB3   -1.2860   2.5590   1.5490 HC   1 NLE    0.096311
   8 CG    -1.3600   3.7940  -0.2280 2C   1 NLE   -0.152189
   9 HG2   -1.8470   3.7630  -1.2030 HC   1 NLE    0.085311
  10 HG3   -0.2880   3.8500  -0.4220 HC   1 NLE    0.085311
  11 CD    -1.8050   5.0560   0.5250 2C   1 NLE   -0.158189
  12 HD2   -2.8560   4.9750   0.8000 HC   1 NLE    0.079811
  13 HD3   -1.2490   5.1380   1.4590 HC   1 NLE    0.079811
  14 CE    -1.6020   6.3320  -0.2980 CT   1 NLE   -0.210189
  15 HE1   -1.8950   7.2150   0.2720 HC   1 NLE    0.074811
```

```
  16 HE2   -0.5590   6.4580  -0.5890 HC   1 NLE    0.074811
  17 HE3   -2.2040   6.3150  -1.2090 HC   1 NLE    0.074811
  18 C      0.2920   1.0090   0.0910 C    1 NLE    0.61779
  19 O      0.7170   0.4370   1.0960 OD   1 NLE   -0.56322
```

Analogous `mol2` files may be prepared for the N- and C-terminal versions of the amino acid; these have different sets of shared charges for backbone atoms, and their net charges should be +1 or -1, respectively. Their backones also use different atom types; NL is used for for N and HP for HA of N-terminal residues, and O3 is used for O and OXT of C-terminal residues. For these, it is likely easiest to edit the `mol2` file manually.

1.3 Preparing an frcmod File

In addition to the atom types and charges of the artificial amino acid, it is necessary to prepare an `frcmod` file including any bonded and nonbonded parameters not already present in the ff14ipq force field. This may be obtained using AmberTools' `parmchk` program:

```
parmchk -f  mol2 -i  norleucine_edit.mol2 \
        -pf 1    -p  $AMBERHOME/dat/leap/parm/parm14ipq.dat \
        -a  N    -o  frcmod.norleucine
```

The resulting outfile lists the mass, bond, angle, dihedral, improper dihedral, and nonbonded parameters that are not present in the ff14ipq force field originally. For norleucine, this includes the bonds, angles, and dihedrals between carbon atom types C8 and CT; bonds between these atom types are not present in any of the standard amino acids modeled by ff14ipq. The parameters for each missing term are initially set to 0; satisfactory initial values may be copied from those of similar atoms. The missing `C8-CT` bond parameters may be copied from those of `CT-CT` and `C8-C8`, these bonds share the same parameters and thus `C8-CT` may be copied unambiguosly. The missing `C8-C8-CT`, `C8-CT-HC`, and `HC-C8-CT` angle parameters may similarly be copied unambiguously. For the dihedral parameters, the appropriate source is less clear; ff14ipq includes several dihedrals that might be considered similar to the missing `C8-C8-CT-HC` and `HC-C8-CT-HC` dihedrals. Here we will copy the `X -CT-CT-X` as the initial value for each of these; later, we will fit unique parameters to the two new dihedrals. The final frcmod file follows:

```
Non-standard amino acid norleucine parameters
MASS

BOND
C8-CT  310.0000  1.5260  Copied from CT-CT, C8-C8

ANGLE
C8-C8-CT  40.0000  109.50  Copied from CT-CT-CT, C8-C8-C8
C8-CT-HC  50.0000  109.50  Copied from C8-C8-HC, CT-CT-HC
HC-C8-CT  50.0000  109.50  Copied from C8-C8-HC, CT-CT-HC

DIHE
C8-C8-CT-HC  1  0.24558  0.0  3.0  Copied from X -CT-CT-X
HC-C8-CT-HC  1  0.24558  0.0  3.0  Copied from X -CT-CT-X

IMPROPER

NONBON
```

1.4 Preparing a lib File

In order to easily construct systems containing norleucine, we may prepare a lib file using AMBER's `tleap` program:

```
tleap

$ source           leaprc.ff14ipq
$ loadamberparams  frcmod.norleucine
$ NLE  = loadmol2  norleucine_edit.mol2
$ NNLE = loadmol2  norleucine_nt_edit.mol2
$ CNLE = loadmol2  norleucine_ct_edit.mol2
$ check    NLE
$ check   NNLE
$ check   CNLE
$ saveoff  NLE    norleucine.lib
$ saveoff NNLE    norleucine.lib
$ saveoff CNLE    norleucine.lib
$ quit
```

This prepares a lib file including the coordinate, connectivity, atom type, and charge information for nonterminal, N-terminal, and C-terminal norleucine. This lib file may subsequently be used by `tleap` to prepare peptides containing norleucine from only a provided sequence. Before it may be used for this purpose; it is necessary to mark which atoms may be involved in peptide bonds. This information is stored in the connect and residueconnect sections, and may be edited as follows:

```
!entry.CNLE.unit.connect array int
 1
 0

!entry.CNLE.unit.residueconnect table  int c1x  int c2x  int c3x  int c4x  int c5x  int
 1 0 0 0 0 0

!entry.NLE.unit.connect array int
 1
 18

!entry.NLE.unit.residueconnect table  int c1x  int c2x  int c3x  int c4x  int c5x  int c
 1 18 0 0 0 0

!entry.NNLE.unit.connect array int
 0
 20

!entry.NNLE.unit.residueconnect table  int c1x  int c2x  int c3x  int c4x  int c5x  int
 0 20 0 0 0 0
```

2.0 Constructing Dipeptide Systems

The lib and frcmod files prepared above are sufficent to build systems containing norleucine using `tleap`:

```
tleap

$ source          leaprc.ff14ipq
```

```
$ loadAmberParams frcmod.norleucine
$ loadoff         norleucine.lib
$ nonterminal   = sequence { ACE NLE NME }
$ n_terminal    = sequence { NLE NME }
$ c_terminal    = sequence { ACE NLE }
$ solvatebox      nonterminal TIP4PEWBOX 12 iso
$ solvatebox      n_terminal  TIP4PEWBOX 20 iso
$ solvatebox      c_terminal  TIP4PEWBOX 20 iso
$ saveamberparm   nonterminal norleucine.prm    norleucine.crd
$ saveamberparm   n_terminal  norleucine_nt.prm norleucine_ct.crd
$ saveamberparm   c_terminal  norleucine_ct.prm norleucine_ct.crd
$ quit
```

This solvates a norleucine dipeptide, blocked with acetyl and N-methyl groups, in a cubic box of TIP4P-Ew water with at least 12 A separating the solute and the periodic boundary of the simulation box. It additionally prepares systems omitting each of the caps; these are used to generate conformations of the N- and C- terminal versions of the amino acids. For these charged systems, the boundary is increased to 20 A.

3.0 Generating Solute Conformations

The solvated parmtop and coordinates for nonterminal, N-terminal, and C-terminal norleucine may now be used to generate solute conformations. The systems are minimized, run through an initial equilibration at constant volume, and run through a longer equilibration at constant pressure. These steps may be carried out using `pmemd`. Comments must be removed from the configuration files below before using them.

3.1 Minimization

Since our objective is to obtain diverse solute conformations, there is no need to restrain the solute during equilibration.

```
&cntrl
  imin      = 1,        # Run minimization
  irest     = 0,        # Do not restart calculation from input file
  ntx       = 1,        # Read input coordinates
  ntmin     = 1,        # Run steepest descent, then conjugate gradient
  maxcyc    = 10000,    # Maximum number of minimization cycles
  ncyc      = 500,      # Number of steepest descent cycles
  ntr       = 0,        # Do not apply position restraints
  ntb       = 1,        # Periodic boundary conditions with constant volume
  ntf       = 1,        # Include bonds to hydrogen in force calculation
  ntc       = 1,        # Do not use SHAKE to restrain bonds to hydrogen
  cut       = 10.0,     # Nonbonded cutoff (A)
  ntpr      = 1,        # Energy log output interval (timesteps)
  ntxo      = 2,        # Output restart file in NetCDF binary format
  ntwr      = 10000,    # Restart file output interval (timesteps)
  ioutfm    = 1,        # Output trajectory in NetCDF binary format
  ntwx      = 10000,    # Trajectory output interval (timesteps)
  iwrap     = 1,        # Write coordinates wrapped
&end
```

3.2 Temperature Equilibration

In order to allow the solute to sample a diverse set of conformations, the simulations are run at 450 K.

```
&cntrl
  imin      = 0,        # Run molecular dynamics
  irest     = 0,        # Do not restart calculation from input file
  ntx       = 1,        # Read input coordinates
  ig        = -1,       # Use random seed from current time
  dt        = 0.002,    # Timestep (ps)
  nstlim    = 10000,    # Simulation duration (timesteps)
  nscm      = 500,      # Center of mass motion removal interval (timesteps)
  ntr       = 0,        # Do not apply position restraints
  ntb       = 1,        # Periodic boundary conditions with constant volume
  ntp       = 0,        # Disable barostat
  ntt       = 3,        # Langevin thermostat
  tempi     = 450.0,    # Initialize velocities from Maxwellian distribution
  temp0     = 450.0,    # System temperature (K)
  gamma_ln  = 1.0,      # Langevin collision frequency (1 / tau) (ps-1)
  ntf       = 2,        # Exclude bonds to hydrogen from force calculation
  ntc       = 2,        # Constrain bonds to hydrogen using SHAKE
  cut       = 10.0,     # Nonbonded cutoff (A)
  ntpr      = 500,      # Energy log output interval (timesteps)
  ntxo      = 2,        # Output restart file in NetCDF binary format
  ntwr      = 10000,    # Restart file output interval (timesteps)
  ioutfm    = 1,        # Output trajectory in NetCDF binary format
  ntwx      = 500,      # Trajectory output interval (timesteps)
  iwrap     = 1,        # Write coordinates wrapped
&end
```

3.3 Volume Equilibration

```
&cntrl
  imin      = 0,        # Run molecular dynamics
  irest     = 1,        # Restart calculation from input file
  ntx       = 5,        # Read input coordinates, velocities, and box
  ig        = -1,       # Use random seed from current date and time
  dt        = 0.002,    # Timestep (ps)
  nstlim    = 500000,   # Simulation duration (timesteps)
  nscm      = 500,      # Center of mass motion removal interval (timesteps)
  ntr       = 0,        # Do not apply position restraints
  ntb       = 2,        # Periodic boundary conditions with constant pressure
  ntp       = 1,        # Constant pressure with isotropic scaling
  barostat  = 2,        # Monte Carlo barostat
  pres0     = 1.0,      # System pressure (bar)
  mcbarint  = 100,      # Number of steps between volume change attempts
  comp      = 44.6,     # Compressibility (1e-6 bar-1)
  taup      = 1.0,      # Barostat time constant (ps)
  ntt       = 3,        # Langevin thermostat
  temp0     = 450.0,    # System temperature (K)
  gamma_ln  = 1.0,      # Langevin collision frequency (1 / tau) (ps-1)
  ntf       = 2,        # Exclude bonds to hydrogen from force calculation
  ntc       = 2,        # Constrain bonds to hydrogen using SHAKE
  cut       = 10.0,     # Nonbonded cutoff (A)
  ntpr      = 500,      # Energy log output interval (timesteps)
  ntxo      = 2,        # Output restart file in NetCDF binary format
  ntwr      = 500000,   # Restart file output interval (timesteps)
  ioutfm    = 1,        # Output trajectory in NetCDF binary format
  ntwx      = 500,      # Trajectory output interval (timesteps)
  iwrap     = 1,        # Write coordinates wrapped
&end
```

3.4 Solute Conformation Generation

A longer simulation from which a series of different conformations are saved may now be run. From this 10 ns simulation, separate restart files are written every 500 ps, yielding a total of 20 different conformations to be used for charge fitting.

```
&cntrl
  imin      = 0,        # Run molecular dynamics
  irest     = 1,        # Restart calculation from input file
  ntx       = 5,        # Read input coordinates, velocities, and box
  ig        = -1,       # Use random seed from current date and time
  dt        = 0.002,    # Timestep (ps)
  nstlim    = 5000000,  # Simulation duration (timesteps)
  nscm      = 500,      # Center of mass motion removal interval (timesteps)
  ntr       = 0,        # Do not apply position restraints
  ntb       = 2,        # Periodic boundary conditions with constant pressure
  ntp       = 1,        # Constant pressure with isotropic scaling
  barostat  = 2,        # Monte Carlo barostat
  pres0     = 1.0,      # System pressure (bar)
  mcbarint  = 100,      # Number of steps between volume change attempts
  comp      = 44.6,     # Compressibility (1e-6 bar-1)
  taup      = 1.0,      # Barostat time constant (ps)
```

```
  ntt       = 3,         # Langevin thermostat
  temp0     = 450.0,     # System temperature (K)
  gamma_ln  = 1.0,       # Langevin collision frequency (1 / tau) (ps-1)
  ntf       = 2,         # Exclude bonds to hydrogen from force calculation
  ntc       = 2,         # Constrain bonds to hydrogen using SHAKE
  cut       = 10.0,      # Nonbonded cutoff (A)
  ntpr      = 500,       # Energy log output interval (timesteps)
  ntxo      = 2,         # Output restart file in NetCDF binary format
  ntwr      = -250000,   # Restart file output interval (timesteps)
  ioutfm    = 1,         # Output trajectory in NetCDF binary format
  ntwx      = 500,       # Trajectory output interval (timesteps)
  iwrap     = 1,         # Write coordinates wrapped
&end
```

4.0 Estimating the Solvent Reaction Field Potential and Performing Quantum Calculations

For each of the three systems, each of the 20 conformations may now be re-minimized using `pmemd`, this time using 10 kcal mol1 A-2 restraints on the solute in order to retain its conformation. In order to be able to transfer the coordinates to `mdgx`, the minimized restart file is output in ASCII format rather than NetCDF.

4.1 Minimization

```
&cntrl
  imin      = 1,         # Run minimization
  irest     = 0,         # Do not restart calculation from input file
  ntx       = 1,         # Read input coordinates
  ntmin     = 1,         # Run steepest descent, then conjugate gradient
  maxcyc    = 10000,     # Maximum number of minimization cycles
  ncyc      = 500,       # Number of steepest descent cycles
  ntr       = 1,         # Apply position restraints
  restraintmask = ':1-3' # Restrain selected atoms
  restraint_wt  = 10.0,  # Position restraint weight (kcal mol-1 A-2)
  ntb       = 1,         # Periodic boundary conditions with constant volume
  ntf       = 1,         # Include bonds to hydrogen in force calculation
  ntc       = 1,         # Do not use SHAKE to restrain bonds to hydrogen
  cut       = 10.0,      # Nonbonded cutoff (A)
  ntpr      = 1,         # Energy log output interval (timesteps)
  ntxo      = 1,         # Output restart file in ASCII text format
  ntwr      = 10000,     # Restart file output interval (timesteps)
  ioutfm    = 1,         # Output trajectory in NetCDF binary format
  ntwx      = 10000,     # Trajectory output interval (timesteps)
  iwrap     = 1,         # Write coordinates wrapped
&end
```

4.2 IPolQ

The minimized structures may now be input to the IPolQ module of `mdgx`; this runs molecular dynamics with the solute atoms fixed in order to estimate the solvent reaction field potential (SRFP) around the solute, and subsequently runs quantum calculations both with and without it, to be used for subsequent charge fitting. This is carried out at 298 K, the temperature of paramaterization of the force field. The quantum calculations may be carried out using either Gaussian or Orca, in this tutorial Orca will be used Running the calculations in parallel using Orca presents a challenge, in that Orca requires OpenMPI for parallelization, but OpenMPI does not allow itself to be run by another OpenMPI process. `mdgx` provides the setting `prepqm` to work around this limitation; this allows shell commands to be run prior to starting Orca. `mdgx` may therfore be run using MPICH, e.g. `/path/to/mpich/bin/mpirun -np 8 mdgx.MPI`, and the `prepqm` setting used to add OpenMPI to `$PATH` and `$LD_LIBRARY_PATH` before starting Orca. Note also that the full path to the Orca executables must be provided; the main `orca` executable uses this path to find other orca executables.

```
&cntrl
  imin      = 0         # Run molecular dynamics
  irest     = 0         # Do not restart calculation from input file
  dt        = 0.002     # Timestep (ps)
  nstlim    = 250000    # Simulation duration (timesteps)
  ntp       = 1         # Constant pressure with isotropic scaling
  barostat  = 2         # Monte Carlo barostat
  pres0     = 1.0       # System pressure (bar)
  mccomp    = 0.002     # Scale of volume change attempts (proportion)
  mcbarint  = 100       # Number of steps between volume change attempts
  ntt       = 3         # Langevin thermostat
  tempi     = 298.0     # Initial system temperature (K)
  temp0     = 298.0     # System temperature (K)
  gamma_ln  = 1.0       # Langevin collision frequency (1 / tau) (ps-1)
  rigidbond = 1         # Constrain bonds to hydrogen using RATTLE
  rigidwat  = 1         # Constrain water bonds to hydrogen using
  es_cutoff  = 10.0     # Electrostatic direct-space cutoff (A)
  vdw_cutoff = 10.0     # van der Waals cutoff (A)
  ntpr      = 500       # Energy log output interval (timesteps)
  ntxo      = 1,        # Output restart file in ASCII text format
  ntwr      = 250000    # Restart file output interval (timesteps)
  ioutfm    = 1,        # Output trajectory in NetCDF binary format
  ntwx      = 500       # Trajectory output interval (timesteps)
  iwrap     = 1,        # Write coordinates wrapped
&end
&ipolq
  scrdir    = /path/to/scratch  # Scratch directory
  solute    = ':1-3'    # Solute atom selection
  ntqs      = 1000      # Rate of charge density sampling
  nqframe   = 200       # Number of frames used to compose the SRFP
  nsteqlim  = 50000     # Number of equilibration steps
  nblock    = 4         # Number of blocks for convergence estimation
  modq      = '@H1'   0.6295    # Charge modifications to be applied to solvent
  modq      = '@H2'   0.6295    #   atoms; in the iPolQ protocol, it is
  modq      = '@EPW' -1.2590    #   appropriate to hyper-polarize solvent
                                #   molecules in the solvent reaction field
                                #   potential calculation; the dipole of
                                #   TIP4P-Ew is therefore increased by an amount
                                #   equal to the model's original dipole - 1.85,
                                #   the experimental dipole of water in vacuum
```

```
  nqshell   = 3         # Number of shells of charges placed around the system
                        #   in order to approximate the solvent reaction field
                        #   potential in the confines of an isolated system
  nqphpt    = 100       # Number of charges placed on each shell around each
                        #   atom in the system; charges are placed equidistant
                        #   on a sphere around each atom; and those charges that
                        #   fall within the spheres of other atoms are removed
  qshell1   = 5.0       # Distance at which to locate first shell of surface
                        #   charges; within this cutoff charges are collected
                        #   explicitly from the simulation's solvent atoms
#  qshell2  = Default?  # Distance at which to locate second shell of surface
#                       #   charges
#  qhsell3  = Default?  # Distance at which to locate third shell of surface
#                       #   charges
  minqt     = 0.01      # Stiffness of harmonic restraint by which to restrain
                        #   fitted shell charges to 0
  nvshell   = 3         # Number of shells of points around the system at which
                        #   the exact solvent reaction field potential due to
                        #   infinite electrostatics will be calculated
  nvphpt    = 20        # Number of points on each shell around each atom in the
#                       #   system; points are placed equidistant on a sphere
#                       #   around each atom; and those charges that fall within
#                       #   the spheres of other atoms are removed
#  vhsell1  = Default?  # Distance at which to locate first shell of points at
#                       #   which to calculate the exact solvent reaction field
#                       #   potential
#  vshell2  = Default?  # Distance at which to locate second shell
#  vshell3  = Default?  # Distance at which to locate third shell
  qmprog    = orca      # Program to use for QM calculations
  qmpath    = /path/to/orca     # Path to QM executable
  prepqm    = "PATH=/path/to/openmpi/bin:$PATH
  prepqm    = "LD_LIBRARY_PATH=/path/to/openmpi/lib:$LD_LIBRARY_PATH"
                        # Shell command(s) to run prior to QM calculation
  maxcore   = 6144      # Maximum memory available to QM program (MB)
  qmlev     = MP2       # Level of quantum theory to use
  basis     = cc-pvTZ   # Basis set
  uvpath    = /path/to/orca_vpot    # Path to electrostatic potential evaluation
                                    # executable
#  unx      = Default?  # Number of grid points on which to evaluate
#                       #   electrostatic potential in x direction
#  uny      = Default?  # Number of grid points on which to evaluate
#                       #   electrostatic potential in y direction
#  unz      = Default?  # Number of grid points on which to evaluate
#                       #   electrostatic potential in z direction
#  uhx      = Default?  # Grid spacing in x direction
#  uhy      = Default?  # Grid spacing in y direction
#  uhz      = Default?  # Grid spacing in z direction
  verbose   = 1         # Verbose output
  qmcomm    = qm_input  # Basename of QM input file
  qmresult  = qm_output # Basename of QM output file
  rqminp    = 1         # Retain QM input files after run
  rqmchk    = 1         # Retain QM checkpoint files after run
  rqmout    = 1         # Retain QM output files after run
  rcloud    = 1         # Retain solvent charge density cloud file after run
  grid      = grid_output   # Base name of electrostatic potential grid file
```

```
  ptqfi     = srfp_output   # Name of point charges file referenced for solvent
                            #   reaction field potential included in
                            #   condensed-phase calculation
&end
```

5.0 Fitting Charges

Charges may now be fit separately for the nonterminal, N-terminal, and C-terminal forms of norleucine, using the 20 pairs of quantum calculations run for each. The fitq module of `mdgx` requires a partmtop file containing only the solute atoms, which may be prepared using AmberTools' `parmed.py`

```
parmed.py norleucine.prm

$ strip :WAT
$ parmout norleucine_solute.prm
$ go
```

Several restrictions must be applied during charge fitting. First, the charges of the ACE and NME blocking groups should maintin their ff14ipq values, as should those of the N, H, C, O (and H1, H2, H3 and OXT) backbone atoms; this may be done using the `sumq` setting. Second, the charges of like atoms should be equal; this may be done using the `equalq` setting. Finally, the charges of buried atoms should be restrained; this may be done using the `minq` setting.

```
&fitq
  # Input vacuum and solvent reacion field potential quantum calculations
  #   from IPolQ
  ipolq    0000.vacu  0000.solv  norleucine_solute.prm  1.0
  ipolq    0001.vacu  0001.solv  norleucine_solute.prm  1.0
  ipolq    0002.vacu  0002.solv  norleucine_solute.prm  1.0
  ipolq    0003.vacu  0003.solv  norleucine_solute.prm  1.0
  ipolq    0004.vacu  0004.solv  norleucine_solute.prm  1.0
  ipolq    0005.vacu  0005.solv  norleucine_solute.prm  1.0
  ipolq    0006.vacu  0006.solv  norleucine_solute.prm  1.0
  ipolq    0007.vacu  0007.solv  norleucine_solute.prm  1.0
  ipolq    0008.vacu  0008.solv  norleucine_solute.prm  1.0
  ipolq    0009.vacu  0009.solv  norleucine_solute.prm  1.0
  ipolq    0010.vacu  0010.solv  norleucine_solute.prm  1.0
  ipolq    0011.vacu  0011.solv  norleucine_solute.prm  1.0
  ipolq    0012.vacu  0012.solv  norleucine_solute.prm  1.0
  ipolq    0013.vacu  0013.solv  norleucine_solute.prm  1.0
  ipolq    0014.vacu  0014.solv  norleucine_solute.prm  1.0
  ipolq    0015.vacu  0015.solv  norleucine_solute.prm  1.0
  ipolq    0016.vacu  0016.solv  norleucine_solute.prm  1.0
  ipolq    0017.vacu  0017.solv  norleucine_solute.prm  1.0
  ipolq    0018.vacu  0018.solv  norleucine_solute.prm  1.0
  ipolq    0019.vacu  0019.solv  norleucine_solute.prm  1.0

  # Lock charges for blocking groups and backbone atoms to their previously-fit
  #   ff14ipq values
  sumq    ':ACE & @HH31'   0.017950
  sumq    ':ACE & @CH3'   -0.013150
  sumq    ':ACE & @HH32'   0.017950
  sumq    ':ACE & @HH33'   0.017950
```

```
  sumq    ':ACE & @C'      0.520730
  sumq    ':ACE & @O'     -0.561430
  sumq    ':NLE & @N'     -0.499980
  sumq    ':NLE & @H'      0.318250
  sumq    ':NLE & @C'      0.617790
  sumq    ':NLE & @O'     -0.563220
  sumq    ':NME & @N'     -0.558840
  sumq    ':NME & @H'      0.341750
  sumq    ':NME & @CH3'   -0.011960
  sumq    ':NME & @HH31'   0.076350

  # Constrain charges of equivalent atoms to be equal
  equalq  ':NLE & @HB2,HB3'
  equalq  ':NLE & @HG2,HG3'
  equalq  ':NLE & @HD2,HD3'
  equalq  ':NLE & @HE1,HE2,HE3'

  # Restrain charges of buried atoms
  minq    ':NLE & @CE'
  minqwt  1.0e-2        # Force constant by which to restrain charges

  nfpt    3750          # Number of fitting points to select from each
                        #   electrostatic potential grid
  flim    0.39          # Minimum proximity of any two points in fit
  psig    3.16435       # Lennard-Jones sigma of solvent probe
  peps    0.16275       # Lennard-Jones epsilon of solvent probe
  pnrg    0.0           # Maximum Lennard-Jones energy of solvent probe at which
                        # a point will qualify for inclusion in the fit
  maxmem  8GB           # Maximum memory available
  verbose 1             # Verbose output
&end
```

5.1 Iteration