

Multi-Conformer Continuum Electrostatics (MCCE)

Gunner Lab at the City College of New York

[\[Home\]](#) [\[Manual\]](#) [\[Tools\]](#) [\[About Us\]](#)

MCCE Manual (in preparation)

Table of Contents

1. Installation

1.1. What is MCCE?

1.2. What is in the MCCE program package?

1.3. Installation

2. Quick Start

2.1. Working directory and control file "run.prm"

2.2. Submitting and monitoring mcce job, a simple example

3. Understanding MCCE program

3.1. File flow chart of MCCE program

3.2. Four steps of the mcce program

3.2.1. Program start, initialization

3.2.2. Step 1, Formatting pdb file

3.2.3. Step 2, Making rotamers

3.2.4. Step 3, Calculating the energy lookup table

3.2.5. Step 3, Simulating pH or Eh titration with Monte Carlo sampling

3.3. Customizing control file - line by line remarks of "run.prm"

3.3.1. The most modified section

3.3.2. The less modified entries

3.3.3. Customizing step 1, preformatting pdb file

3.3.4. Customizing step 2, rotamer making

3.3.5. Customizing step 3, energy lookup table calculation

3.3.6. Customizing step 4, Monte Carlo sampling

3.3.7. Misceleneous entries

3.4 Tricks of using "run.prm"

3.4.1 Detecting ligand

3.4.2 Updating head3.lst

3.5. Making parameters for a new cofactor

3.5.1. Make a tpl file

3.5.2. Adding atomic charges into the tpl files

3.5.3 Get pKa, Em, charges and calculated energies

3.6. Running MCCE on a computer cluster

4. Understanding MCCE results

4.1. MCCE Physics

4.2. MCCE program

4.3. Calculating pKa

4.4. Calculating Em

4.5. Proton uptake upon reduction

4.6. The controlling factors of pKa and Em

4.7. Free energy of ionizing a residue at specific pH and Eh

4.8. MCCE tools

5. Trouble Shooting

Appendix: Input and Output files of MCCE

1. Installation

1.1. What is MCCE?

MCCE (Multi-Conformation Continuum Electrostatics) is a biophysics simulation program combining continuum electrostatics and molecular mechanics. In this program, the protein side chain motions are simulated explicitly while the dielectric effect of solvent and bulk protein material is modeled by continuum electrostatics. MCCE can study:

1. Residue ionization and side chain conformation as a function of solution pH or Eh;
2. Residue pKa and redox cofactor E_m ;
3. Side chain conformer packing;
4. Distribution and orientation of buried waters;
5. Stoichiometry and pH dependence of proton coupled electron transfer;
6. Changes in ionization and side chain conformation at different stages of a reaction or with different ligands bound.

1.2. What is in the MCCE program package?

MCCE 1.0 includes "mcce" main program, Poisson-Boltzmann equation solver "DelPhi" developed in Barry Honig's Lab (You need to get separate license from Barry Honig's Lab to use DelPhi at <http://honiglab.cpmc.columbia.edu/delphi>), MCCE tool kits, and parameter directories.

1.3. Installation

After you get file "mcce1.0.tar.gz", run

```
tar -zxvf mcce-1.0.tar.gz
cd mcce-1.0
./mcce_install
```

In the installation process, you will be asked questions about license issues and the desired location of the program.

The MCCE program requires library "gdbm" and "glibc2.2" or above on your system, and the installation will detect if your system satisfies the requirement.

After a successful installation, you may want to add mcce bin directory into your environment variable \$PATH. Ask your system administrator if you need help on this.

2. Quick Start

2.1. Working directory and control file "run.prm"

MCCE is a command line program. MCCE expects a control file named "run.prm" and will write out results as well as temporary files in the directory where the mcce command is invoked. This directory is called "working directory". Usually the "working directory" is set up to hold a pdb file and a "run.prm" file to start the calculation. We recommend that each protein has it's own directory.

The control file gives instruction to the mcce program. This file must be named "run.prm" and placed in the working directory. Two rules apply to this file: 1) Any line starting with "#" is a comment line and will be ignored; 2) If a line is not a comment line, MCCE interprets the line as a key-value pair in which the string in parentheses is the key and the first string of this line is the value. Here is an example of the most often customized lines of the control file "run.prm":

```
=====
Most modified entries
-----
Input and Output:
prot.pdb                               (INPDB)

Steps:
f step 1: pre-run, pdb->mcce pdb       (DO_PREMCCE)
f step 2: make rotatmers               (DO_ROTAMERS)
f step 3: do energy calculations       (DO_ENERGY)
f step 4: monte carlo sampling         (DO_MONTE)
=====

=====
Less modified entries
-----
8.0 Protein dielectric constant for DelPhi (EPSILON_PROT)
/usr/local/mcce-1.0/extra08.tpl          (EXTRA)
/usr/local/mcce-1.0/name.txt MCCE renaming rule. (RENAME_RULES)

ph "ph" for pH titration, "eh" for eh titration (TITR_TYPE)
0.0 Initial pH                             (TITR_PH0)
1.0 pH interval                            (TITR_PHD)
0.0 Initial Eh                             (TITR_EH0)
1.0 pH interval                            (TITR_PHD)
0.0 Initial Eh                             (TITR_EH0)
30.0 Eh interval in mV                     (TITR_EHD)
15 Number of titration points              (TITR_STEPS)
=====
/usr/local/mcce-1.0                       (MCCE_HOME)
```

In the most often modified section, the input structure must be specified by the line "(INPDB)". Change the name prot.pdb to the name of your file. The file can be in a distant directory with the full path in the name.

There are 4 steps in a MCCE run. You can run them all at once or stop between steps. Each step is dependent on output of the previous ones written into the working directory (see Figure 1, file flow of mcce, which shows needed input and output files for each step). To run a step of MCCE, mark its flag "t".

In the less often modified section, you may specify the dielectric constant of the protein in line "(EPSILON_PROT)". Currently, dielectric constants 8 and 4 are supported, but it is user expandable. To use a different dielectric constant, one needs to find reference reaction field energy for residues at the same dielectric constant and put them into the parameter files. Please ask us and we will add the instruction to the manual. Normally dielectric constant 8 is good for small soluble proteins (< 200 residues) and dielectric constant 4 is good for big transmembrane proteins (>200 residues). The solvent is always assigned a dielectric constant of 80.

The line "(EXTRA)" points to a file with offsets for each residue type calibrated by comparing about 600 calculated pKas and experimental pKas. The origin of the system error is possibly from inaccurate solution pKa of residues, different solvent entropy effect and different polarization of atomic charges between the reference system water and protein material. Leave this line empty if you don't want to use this. This file is a supplemental parameter file to the parameter directory. This file follows the same format as files in the parameter directory and entries in this file have higher priority than those in the parameter directory. This provides means to test a new parameter file before depositing it to the permanent parameter directory.

The file specified in "(RENAME RULES)" allows you to treat individual residues differently. It is sometimes necessary to split a cofactor into several ionizable groups or combine ligand residues with cofactor to be one group. For example, you may want to treat a His ligand to a heme differently than other His or treat the propionic acids on a heme as separate titratable groups. This can be achieved by renaming the atom name, residue name and sequence number of these residues. A sample rename file "name.txt" is in the MCCE distribution directory. . You will need to change the residue names and create a new tpl file for this special residue type.

There are 4 steps in a MCCE run. You can run any of them, but the next step is dependent on the output of the previous step. To run through MCCE, mark the flags for these steps as "t". The Monte Carlo Sampling can do either a pH titration or a redox potential (Eh) titration. The lines in the sample file "run.prm" configure a pH titration from pH=0 to pH=14 with a step of 1 pH unit.

The line "(MCCE_HOME)" defines the location of mcce program and the parameter directories. If you have more than one distribution of mcce programs, you may want to switch between the programs and the parameter directories. For now, the only use of this line is to compose the parameter directory. For example, if this line points to "/usr/local/mcce-1.0" and "(EPSILON_PROT)" gives dielectric constant 4, then the parameter directory is "/usr/local/mcce-1.0/param04".

There are 3 sample "run.prm" files included in the MCCE directory. They are different in the detail level of making rotamers. "run.prm.quick" is suitable for a quick and dirty pKa calculation which is done with minimal number of rotamers. "run.prm.default" provides moderate number of rotamers and predict more accurate pKa values than "run.prm.quick". "run.prm.full" will produce extensive rotamers and is good for exploring side chain motions. Here is a comparison of computing time and calculated pKas of Hen lysozyme (pdb 4lzt):

Table 1. Comparison of runs on lysozyme with the three run.prm presets. When comparing these numbers with experimental pKa values, bigger errors are often seen for the binding sites and functionally important residues marked by *. These residues often show abnormal pKa values which are hard to calculation. An error about 1 pH unit on these sites is often seen.

| Presets of run.prm | run.prm=run.prm.quick | run.prm=run.prm.default | run.prm=run.prm.full |
|--------------------------------|------------------------------|--------------------------------|-----------------------------|
| Number of conformers | 281 | 1421 | 1826 |
| Step 1 - check protein | 1 second | 1 second | 1 second |
| Step 2 - make rotamers | 0.05 minutes | 10 minutes | 19 minutes |
| Step 3 - generate lookup table | 20 minutes | 303 minutes | 456 minutes |
| Step 4 - run Monte Carlo | 1 minute | 47 minutes | 66 minutes |
| Overall RMSD of error | 0.75 | 0.72 | 1.06 |
| | Exp. | | |

| | pKa | | | |
|-----------|------|-------|-------|-------|
| LYS+0001 | 10.8 | 10 | 10.02 | 10.44 |
| GLU-0007 | 2.85 | 3.57 | 3.52 | 3.33 |
| LYS+0013 | 10.5 | 11.44 | 11.46 | 10.99 |
| ASP-0018 | 2.66 | 2.94 | 2.74 | 2.88 |
| TYR-0020 | 10.3 | 11.13 | 10.44 | 11.08 |
| TYR-0023 | 9.8 | 10.14 | 10.18 | 10.61 |
| LYS+0033 | 10.4 | 10.07 | 9.97 | 9.82 |
| GLU-0035* | 6.2 | 5.21 | 5.11 | 5.33 |
| ASP-0048 | 1.6 | 2.56 | 3.09 | 3.83 |
| ASP-0052* | 3.68 | 2.82 | 3.68 | 4.83 |
| ASP-0066* | 0.9 | 2.58 | 2.1 | 3.08 |
| ASP-0087 | 2.07 | 2.46 | 2.41 | 3.65 |
| LYS+0096 | 10.8 | 11.22 | 11.55 | 11.19 |
| LYS+0097 | 10.3 | 11.12 | 11.24 | 10.43 |
| ASP-0101 | 4.08 | 4.26 | 4.25 | 5.34 |
| LYS+0116 | 10.2 | 9.73 | 9.71 | 9.79 |
| ASP-0119 | 3.2 | 3.62 | 3.58 | 4.3 |
| CTR-0129 | 2.75 | 2.44 | 2.46 | 3.54 |

2.2. Submitting and monitoring mcce job, a simple example

Here is an example of calculating pKa values for Hen egg lysozyme. In this example, mcce was installed under /usr/local/mcce-1.0/, and the directory /usr/local/mcce-1.0/bin/ is in my executable searching path. First make a directory for this set of run. You get the structure file 4LZT from the rscb data site with command "getpdb" included in MCCE package:

```
jmao@metis jmao]$ mkdir 4lzt
[jmao@metis jmao]$ cd 4lzt
[jmao@metis 4lzt]$ getpdb 4lzt
Inquiring the remote file 4LZT.pdb ...
Saving as 4LZT.pdb ...
Download completed.
[jmao@metis 4lzt]$
```

Then copy a "run.prm" file to this directory and edit it:

```
[jmao@metis 4lzt]$ cp /usr/local/mcce-1.0/run.prm.quick run.prm
```

You will need to edit this "run.prm" file. In the editor, make first string of line (INPDB) "4LZT.pdb", and turn flags for all 4 steps to "t". You may want to run one step at a time and look at the output and run.log messages for each step.

The Most modified section will look like this:

```
=====
Most modified entries
-----
Input and Output:
4LZT.pdb ( INPDB)

Steps:
t step 1: pre-run, pdb->mcce pdb (DO_PREMCCE)
t step 2: make rotatmers (DO_ROTAMERS)
t step 3: do energy calculations (DO_ENERGY)
t step 4: monte carlo sampling (DO_MONTE)
=====
```

MCCE can be invoked simply by running the executable file "mcce". As the program will take hours to days to finish, you may want to run the program at the background and redirect the output message to a log file using command "mcce > run.log &".

```
[jmao@metis 4lzt]$ mcce > run.log&
[1]8533
```

While running, MCCE writes out the progress, warning messages and error messages directed here into file "run.log". Check mcce run time messages and serious errors will appear in it. The "debug.log" and "progress.log" provide extra warning messages and progress report, which can be safely ignored most time.

In running 4LZT I noticed a line in the run.log file:

```
Error! premcce_confname(): Can't get conformer list of this residue NO3
```

This means a cofactor NO3 was not recognized by mcce. The program continues anyway and treats all 4 atoms of NO3 with 0 charge and radii of 1.7 Angstroms. You may want to stop the program and make a TPL (residue parameter file) file, but we will ignore the error. If NO3 is on the surface, it won't cause much error in pKa because of 1) the screening effect of the solvent and 2) Poisson-Boltzmann equation will take care of the salt effect if this surface NO3 is stripped off like surface water by the mcce program.

After step 1, the file "step1_out.pdb" is a mcce format pbd file with terminal residues split off to form an independently titratable NTR and CTR, groups renamed based in file "name.txt", and surface water and salt molecules with 10% solvent accessibility stripped off. Undefined cofactors will be identified as made up of non-charged atoms. A temporary parameter file "new.tpl" will be written out. This can help you start to make a tpl file for new groups. The file "acc.res" lists the solvent accessibility of residues and the file "acc.atm" lists the solvent accessibility of atoms. No conformers are made for groups with more than 30% surface accessibility in the configuration of "run.prm.default".

After step 2, you will get a multi-conformation pdb file "step2_out.pdb", which is the only file used by step 3. The rotamer making statistics are dynamically updated in file "rot_stat".

The DelPhi runs in step 3 are the most time consuming step in mcce. If you look at the bottom of run.log when DelPhi is running it will look like this:

```
[jmao@metis 4lzt]$ tail run.log

Creating connectivity table...
Done

Preparing DelPhi runs ...
  3 focusing runs required for this protein.
Done

Computing pairwise from conformer 1 to 281 of 281 total conformers
  see progress.log for progress...
```

So, we take a look at the additional progress file "progress.log":

```
[jmao@metis 4lzt]$ tail progress.log
Doing pairwise 72 of 281 conformers. 5 seconds
Doing pairwise 73 of 281 conformers. 2 seconds
Doing pairwise 74 of 281 conformers. 4 seconds
Doing pairwise 75 of 281 conformers. 2 seconds
Doing pairwise 76 of 281 conformers. 3 seconds
Doing pairwise 77 of 281 conformers. 3 seconds
Doing pairwise 78 of 281 conformers. 4 seconds
Doing pairwise 79 of 281 conformers. 4 seconds
Doing pairwise 80 of 281 conformers. 5 seconds
Doing pairwise 81 of 281 conformers. [jmao@4lzt]$
```

This progress is reported in "progress.log" and it can be used to estimate how long step 3 will take. A pairwise calculation and a reaction field energy calculation will be performed for each conformer. So step 3 will take about 4 seconds*281 conformers*2 DelPhi's= 2248 seconds or 37 minutes. The output directory of step 3 is "energies" and the output file is "head3.lst". Both are needed as input of step 4.

Step 4 is Monte Carlo sampling, which simulates a pH or Eh titration. It writes the progress to file "mc_out". The occupancy of each conformer at each pH is in fort.38. The net charge of each ionizable residue is in "sum_crg.out". The pKa or Ems are in "pK.out". These are obtained by non-linear fitting of the titration curves in file "sum_crg.out".

One reason to run simulations is to be able to ask WHY. To understand what contributes to the ionization energy of a residue at a specific pH, use the tool "mfe.py". This program approximates the interactions between residues by Mean Field approach. To look at the free energy of the reaction $GLU \rightarrow GLU^- + H^+$ for Glu 58 at pH 5.078, the midpoint of the titration curve of this residue run.

```
[jmao@metis 4lzt]$ mfe.py GLU-_0035_ 5.078 0.2
Residue GLU-_0035_ pKa/Em=5.078
=====
Terms          pH      meV    Kcal
-----
vdw0           0.03    1.98    0.05
vdw1           0.08    4.91    0.12
tors          -0.00   -0.00   -0.00
ebkb          -1.39  -80.41  -1.89
dsol           2.72  157.67   3.71
offset        -0.99  -57.45  -1.35
pH&pK0        -0.33  -19.04  -0.45
Eh&Em0        0.00    0.00    0.00
-TS           0.11    6.35    0.15
residues      -0.15   -8.47   -0.20
*****
TOTAL          0.10    5.54    0.13
*****
ASP_0048_     0.22   13.02    0.31
ASP_0052_     0.95   54.90    1.29
ARG_0112_    -0.29  -16.64   -0.39
ARG_0114_    -0.24  -13.93   -0.33
=====
```

The program mfe.py is bundled with the mcce package. The syntax of the first argument about how to refer to residue should be taken from pK.out. The second argument means the Boltzmann distribution of conformers for all residues at 5.078 will be used. The Last argument says only specific interactions with individual residues over 0.2 kcal/mol will be printed out individually. The result explains the free energy of the reaction $GLU \rightarrow GLU^+$ at pH 5.078, the midpoint of the titration curve of this residue, where reactant and product are equally favored. So it is expected here that the ionization energy in line TOTAL is close to 0. The small error is caused by mean field approximation and this error may indicate the degree of coupled ionization or conformational changes with other residues. The breakdown of ionization energy helps identifying the factors controlling the pKa of the residue. Negative number favors ionization, so ASP52 in this protein shifts up the pKa of GLU35 by 0.95 pH unit.

Listed in the table are other energy terms. All these terms are the difference between the contributions from the disignated term to the ionized conformers and natural conformers. "vdw0" is van der Waals interaction of the side chain of this residue. "vdw1" is the van der Waals interaction of the side chain of this residue to the backbone atoms. tors is the torsion energy of this residue. "ebkb" is the electrostatic interaction of the side chain to the backbone. "dsolv" is the loss of solvation energy. "offset" is a correction for each residue type defined in "extra.tpl". "pH&pK0" is the pH effect, which equals solution pH minus solution pKa of that residue if the residue is an acid or solution pKa of that residue minus solution pH if the residue is a base. "Eh&Em0" is the redox potential effect, which is the current solution potential minus the standard redox potential of that cofactor. "-TS" is entropy effect. As the protonated and deprotonated residue have have different constraints in a protein, the entropy of the two ionization states could be different. "residues" refers to the mean field interaction with residues given their conformations at this pH. This is the term that is pH or Eh dependent.

A mcce web interface analysis tool is provided separately. If it is installed, just point your browser to "<http://localhost/cgi-bin/mcce.py>" (replace localhost by the host name of mcce server if you run mcce on a remote computer). For more information, refer to section 4, Understanding MCCE Results.

3. Understanding MCCE program

3.1. File flow chart of MCCE program

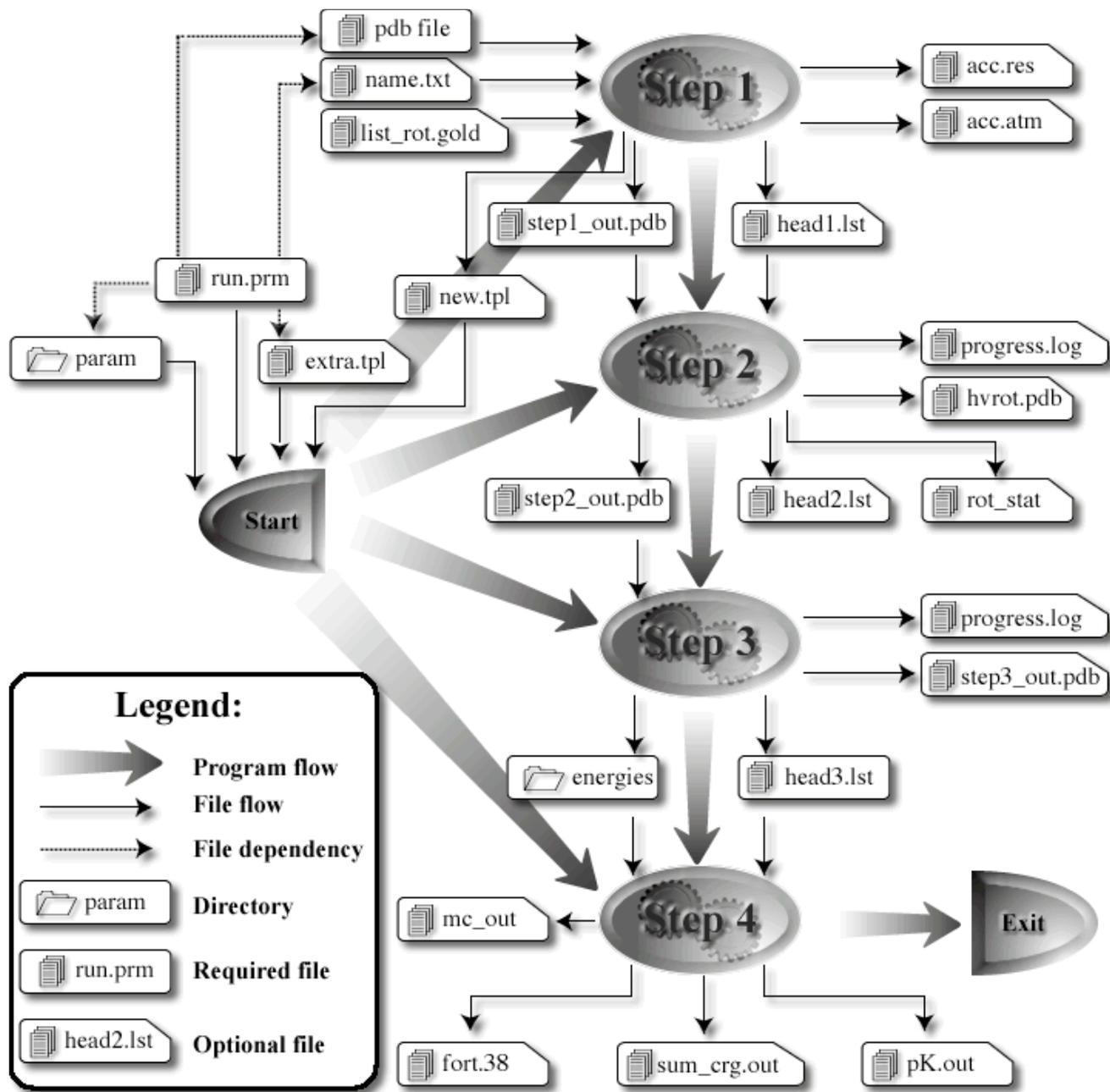


Figure 1. File flow of the mcce program. Files required and written out by the program are illustrated in this chart. Step 1 is file formatting, step 2 is rotamer making, step 3 calculates energy look-up table, and step 4 is Monte Carlo sampling of conformers at each pH and Eh. MCCE program can run any steps providing the prerequisite files exist in the working directory. This file flow chart shows the file dependencies.

3.2. Four steps of the mcce program

There are 4 major steps in a mcce calculation. These 4 steps are connected by a few files. The program is designed to run through without stopping although you can stop the program at each step and edit the files to instruct the next step. Here is the summary of the function of these 4 steps.

3.2.1. Program start, initialization

Input files:

- run.prm: *mcce control file*
- param (specified in "run.prm"): *parameter directory*
- extra.tpl (optional, specified in "run.prm"): *extra parameters*

- new.tpl (optional, created by step 1): *temporary parameter file for unrecognized cofactor*

Output files: None

The initialization reads in control file "run.prm" and loads parameters. It is done everytime mcce is invoked regardless of the steps the program will do.

The control file "run.prm" must be in the working directory where mcce program is called. The lines of this file are interpreted by the mcce program as key-value pairs. The string in parentheses is the key and the first string of the same line is the value. For example the line

```
/usr/local/mcce-1.0                (MCCE_HOME)
```

can be viewed as $\$(MCCE_HOME) = "/usr/local/mcce-1.0"$. Other lines are interpreted in the same way. The resulting key-value pairs define the mcce environment variables, input and output file names, and flags etc.

The mcce program will load parameter files from the parameter directory defined in "run.prm". The way of defining this directory is composed by two variables: $\$(MCCE_HOME) + "/param" + \$(EPSILON_PROT)$. Thus for these lines in the run.prm file:

```
/usr/local/mcce-1.0                (MCCE_HOME)
8.0 Protein dielectric constant for DelPhi (EPSILON_PROT)
```

the path name is `"/usr/local/mcce-1.0/param08"`, in which the dielectric constant is converted to a 2-digit integer. In this directory, mcce program will read in files with extension ".tpl". Files whose names do not end up with ".tpl" will be ignored. This makes it possible to store backup files in the same directory.

The file "extra.tpl" is defined by the variable $\$(EXTRA)$ in "run.prm". This file is optional. If the file exists, mcce reads it in with the same way as the files in the parameter directory. This file has the same format as a regular parameter file. The function of this file is

1. to provide the means to overwrite specific entries in the parameter files of the global parameter directory;
2. to add extra parameters specific to individual proteins such as scaling factors of energy terms;
3. to serve as test parameter file for a new cofactor;

Unlike the parameter directory, this file is intended be customized for individual proteins.

The file "new.tpl" is a parameter file similar to those in the parameter directory except no charges are assigned to any atom. This file is for unrecognized residues or cofactors. In step 1, mcce program treats these cofactors as non-charged atom assemblies and writes out this file for two purposes:

1. This file can be a template of writing the formal parameter file for the unrecognized cofactors;
2. The mcce program will use this temporary parameter file to load unrecognized cofactors when mcce resumes from step 2, 3 and 4.

There is no output file from the program initialization, but the initialization creates a parameter database to hold information from the parameter files and dynamically generated parameters by the program. A temporary image file of the database is found in the working directory as `"~temp.dbm"`. While mcce is running, you can not delete this file or start another mcce job in the same directory.

3.2.2. Step 1, Formatting pdb file

Input files:

- pdb file (specified by "run.prm"): *input structure file in pdb format*
- name.txt (optional, specified by "run.prm") : *residue and atom renaming rule*
- list_rot.gold (optional): *hot residue spot definition*

Output files:

- acc.res: *solvent accessibility of residues*

- acc.atm: *solvent accessibility of atoms*
- new.tpl (not always created): *parameter file template of unrecognized cofactors*
- head1.lst (optionally used by step 2): *summary of rotamer making policy of residues*
- step1_out.pdb (used by step 2): *step 1 output file is in mcce extended pdb format (note: this file has lots of extra characters but can be read as is by many visualization programs)*

Step 1 prepares an extended pdb file suitable to be read into step 2. The input pdb file is in standard pdb format. It can have alternative side chain positions, but mcce can not process alternative backbone positions. Alternative side chains are treated as side chain conformers. When side chain atoms are missing, mcce will complete the side chain atoms at the torsion minimum. In this step several things will happen:

1. With the instructions in the renaming rule file "name.txt", residues and atoms will be renamed so that a cofactor can be split into several independent ionizable groups (for example, heme can be divided into heme and two propionates) and several groups can be combined as one (for example, heme can be grouped with the axial ligands).
2. Step 1 will identify unrecognized cofactors and interpret them as non-charged atom assemblies.
3. This step completes the missing atoms in each known residue (note: If your file has individual residues with missing atoms the tpl file for that residue will be used to add missing atoms).
4. This step calculates the solvent accessible surface (SAS) area and strips off exposed water and salt (HOH, NO3 and SO4). The SAS threshold of stripping off water and salt is defined by \$(H2O_SASCUTOFF).
5. From the hot residue spot file "list_rot.gold" and the rotamer making rules defined in "run.prm", the residue specific rotamer making policy is composed and written to file "head1.lst", which can be used by the step 2.
6. This step identifies geometry clashes between atoms which are not supposed to be bonded.
7. This step extracts N terminus and C terminus of a chain.

The renaming rule file "name.txt" instructs the mcce program to rename atom name, residue name, seque number, and chain ID. Here is several sample lines in this file:

```
# Symbol "*" in the first string is a wildcard that matches any character.
# It means "do not replace" in the second string.
#
# The replace is accumulative in the order of appearing in this file.#
*****HEA*****      *****HEM*****
*****HEC*****      *****HEM*****
*CAA*HEM*****      *****PAA*****      extract PAA from heme
*CBA*HEM*****      *****PAA*****
*CGA*HEM*****      *****PAA*****
#   ATOM      70   CB   ASP   A   12
```

The line started with "#" and the line shorter than 30 characters are comment lines. For other lines, the first 30 characters should be two 14-character strings separated by exactly two spaces, and the rest of the line is comment field. A valid line instructs mcce program to replace string 1 by string 2. The mcce program will match this string with position 13 to 26 of a coordinate line in the input pdb file. The symbol "*" is the wild card that matches any character in strings. The replace action is accumulative and order sensitive. For example, The line

```
HETATM 1683  CAA HEC      1      1.317  -3.987  -1.685  1.00  0.00      C
```

will be renamed to

```
HETATM 1683  CAA HEM      1      1.317  -3.987  -1.685  1.00  0.00      C
```

then

```
HETATM 1683  CAA PAA      1      1.317  -3.987  -1.685  1.00  0.00      C
```

Another file step 1 may use is "list_rot.gold". This file defines "hot spots" in a protein where rotamers will be made with a step of 30 degrees in step 2. If a coordinate line of any atom of a residue is present in this file, this residue and residues within 4 angstroms will be flagged to be "hot spots" in file "head1.lst", which will be used by step 2.

The output files of step 1 "acc.res" and "acc.atm" contain the solvent accessible surfaces of residues and atoms. In "acc.res", both absolute value and percentage of the solvent accessibility are listed.

When unrecognized cofactor is encountered in step 1, a parameter file will be created with name as "new.tpl" and a warning message will be issued. The atom connectivity is guessed and all atoms are assumed to have charge 0. This file can be the

starting point of making a parameter for a new cofactor. If the program is resumed from step 2 or 3, this file will be read in by step 0 as a parameter file so step 2 and 3 can proceed without step 1.

The file "head1.lst" lists the rotamer making policy of the residues. When \$(ROT_SPECIF) in "run.prm" is set to "t", this file will be used as instruction of rotamer making of step 2, otherwise, file "head1.lst" will be ignored. This file can be modified and will be effective if the program resumes from step 2.

The file "step1_out.pdb" is a formatted pdb file, which will be read in by step 2. The mcce extended pdb format contains 3 more fields than standard pdb file. Right after atom coordinates, these three fields are charge, size and rotamer making history.

3.2.3. Step 2, Making rotamers

Input files:

- step1_out.pdb: *input structure file of step 2 in mcce extended pdb format*
- head1.lst (optional) : *rotamer making policy of residues*

Output files:

- progress.log: *progress report file, dynamically updated*
- rot_stat: *rotamer making statistics, dynamically updated*
- hvrot.pdb: *heavy atom (without hydrogen atoms) rotamer pdb file*
- head2.lst (optionally used by step 3): *summary of rotamers made in step 2*
- step2_out.pdb (used by step 3): *step 2 output file with multiple rotamers in extended pdb format*

Step 2 makes and optimizes rotamers from the structure in "step1_out.pdb". In this step, the rotatable bonds (defined in parameter files) of each residue is rotated by the steps defined in "run.prm". Then the self van der Waals (VDW) potential (interaction among atoms of the same side chain excluding 1-2 and 1-3 interactions, and interaction between the side chain and backbone atoms) is calculated. Side chain rotamers with high self VDW potentials are deleted. Then the side chains are optimized with possible hydrogen bond partners. A number of repackings starting from randomized initial structures (one conformer from one residue) are performed to reduce side chain rotamers to those with low energy local packings. Ionization states are then created and protons are placed on side chains. At the end of side chain rotamer optimization, MD simulations are carried out locally to relax the structure.

The input file step1_out.pdb is the only essential file step 2 will use. You can modify this file to add, delete or edit residues without causing problems as long as the file observes mcce extended pdb format. Sometimes a little editing of this file is necessary, for example, the terminal residues are not always correctly identified by the mcce program due to "broken chains" caused by the missing residues in the pdb file.

The file "head1.lst" provides residue specific rotamer making rule. It will be used only when \$(ROT_SPECIF) is set to be "t". The line of this file such as:

```
NTR A0003_ R t 06 S f 0.0 H t 06 M 000
```

is interpreted as "Rotate is true and 6 steps per bond, then swing is false (angle is 0 if any), then Hydroxyl relaxation is true and the maximum number of starting conformers per hydroxyl is 6, and the maximum total number of the conformers is not limited". If you want to investigate a specific site in details, change the step 6 to 12. But making 12 steps for many sites (>30) are not recommended because it is expensive in terms of memory and CPU time.

The file "progress.log" is a dynamically updated progress report file. It lists mainly the repacking progress.

The file "rot_stat" is an important file to review the rotamer making history. This file lists the number of rotamers of each residue. It is easy to tell what residues get rotamers and if the total number of rotamers is manageable (a 2000 conformer structure may need one day to run the next two steps, step 3 and 4).

The file hvrot.pdb is a file at the same format as step1_out.pdb and step2_out.pdb but lacks hydrogen atoms. The main use of this file is to rename it to "step1_out.pdb" and run step 2 again with "swing" rotamers instead of "rotate" rotamers. This provides a way to relax the structure by "swinging" the rotatable bond a little and reevaluate the rotamers. This feature is most for advanced users to calibrate hydrogen bond directed rotamer making algorithm and MD relaxation subroutine.

The file "head2.lst" is a summary of the rotamers made in step 2. This file is not used by step 3.

The file "step2_out.pdb" is in the mcce extended pdb format. It is the single file connecting step 2 and step 3.

3.2.4. Step 3, Calculating the energy lookup table

Input files:

- step2_out.pdb: *input structure file of step 3 in mcce extended pdb format*

Output files:

- progress.log: *progress report file, dynamically updated*
- energies (directory of energy lookup table used by step 4): *opp files, each opp file is pairwise interaction to a conformer*
- head3.lst (used by step 4): *list of conformers and self energy of conformers. It is used by step 4.*
- step3_out.pdb: *step 3 output file with multiple rotamers in extended pdb format*

Step 3 calls PB equation solver, DelPhi, to calculate reaction field energy and electrostatic pairwise interaction. The result is stored as together with van der Waals interactions as one file per conformer. These files have extension "opp" and are located under directory energies. The self-energy terms (not dependent on side chains of other residues) of conformers are listed in file "head3.lst" The progress is dynamically updated in file "progress.log".

The file "progress.log" reports the progress of DelPhi calculation, which can be used to estimate the total time of this step. There will be two parts of DelPhi calculations: the first is pairwise calculation and the second part is the reaction field energy calculation. It takes less time on reaction field energy calculation than on pairwise calculation.

The directory "energies" holds the resulted pairwise interaction lookup table. The first column is the electrostatic pairwise interaction and the second column is van der Waals interaction. The unit of the energy is Kcal/mol.

The file "head3.lst" contains self energy of each conformer and control flags of step 4. The flag is: "t" for fixed occupancy 0 or 1, or "f" for free to sample. The energy unit is Kcal/mol.

The file "step3_out.pdb" is an extended pdb file with multiple conformers. The conformer number is sorted to be continuous and consistent with the conformer numbers in file "head3.lst" and step 4 output file fort.38. This file is identical to "step2_out.pdb" if "step2_out.pdb" is an unmodified file created by step 2.

3.2.5 Step 3, Simulating pH or Eh titration with Monte Carlo sampling

Input files:

- energies (directory): *energy lookup table for pairwise interaction between conformers*
- head3.lst: *self-energy of conformers and Monte Carlo sampling flags of conformers*

Output files:

- mc_out: *progress of Monte Carlo sampling and energy tracing*
- fort.38: *conformer occupancies*
- sum_crg.out: *residue net charges*
- pK.out: *pKa or Em values obtained by titration curve fitting*

Step 4 is a titration simulation by Monte Carlo sampling. The Monte Carlo sampling is performed at specified set of pH/Eh. At each titration point, there will be several (predefined in "run.prm", the default is 6) independent samplings. Each sampling goes through annealing, reducing, and equilibration stages. Statistics of conformer occupancy is only done at equilibration stage. Yifan's Monte Carlo subroutine will check early convergence and quit sampling early to save time. The result is reported as conformer occupancy in file "fort.38", residue net charge in file "sum_crg.out" and fitted pKa/Em values in file "pK.out".

The file "mc_out" is the progress report of Monte Carlo sampling. It contains running energy tracing which can be used to calculate the average E or enthalpy of the system, or verify if the system is trapped at local energy minima. By "grep Sg mc_out", you can find the standard deviation of independent samplings.

3.3. Customizing control file - line by line remarks of "run.prm"

3.3.1. The most modified section

Five entries in this section outline a mcce run. One can specify the input pdb file and steps.

```
prot.pdb (INPDB)
```

This line specifies the input structural file of step 1.

```
f step 1: pre-run, pdb-> mcce pdb (DO_PREMCCE)
```

Step 1 converts regular pdb file to mcce extended pdb format. The flag "t" designates "do this step".

```
f step 2: make rotamers (DO_ROTAMERS)
```

Step 2 makes and optimizes rotamers of the protein. The flag "t" designates "do this step".

```
f step 3: do energy calculations (DO_ENERGY)
```

Step 3 prepares energy look-up table for conformers. The flag "t" designates "do this step".

```
f step 4: monte carlo sampling (DO_MONTE)
```

Step 4 simulates a pH or Eh titration with Monte Carlo sampling. The flag "t" designates "do this step".

3.3.2. The less modified entries

These are additional important controls of the program.

```
8.0 Protein dielectric constant for DelPhi (EPSILON_PROT)
```

Dielectric constant of protein. usually epsilon 8 is good for small soluble protein while 4 is good for big trans-membrane protein (>200 residues). If one wants to use dielectric constant other than 8 and 4, one has to prepare a parameter directory named "param##" where "##" is the dielectric constant as an integer. In this parameter directory, the "RXN" entries of tpl files should be recalibrated for the new dielectric constant.

```
/usr/local/mcce-1.0/extra.tpl (EXTRA)
```

This line points to a file that contains entries to overwrite those in the parameter directory and provides extra controls over the program.

```
/usr/local/mcce-1.0/name.txt MCCE renaming rule. (RENAME_RULES)
```

This is the renaming rule file. It is sometimes necessary to divide a big group into separate ionizable groups, or merge ligands and cofactors as one group. This can be done by renaming the atoms and residues instructed by this file.

```
ph "ph" for pH titration, "eh" for eh titration (TITR_TYPE)
```

The type of titration simulation of step 4, pH or Eh.

```
0.0 Initial pH (TITR_PH0)
```

The starting pH of titration simulation, or the pH of an Eh titration.

```
1.0 pH interval (TITR_PHD)
```

The increment of pH titration (size of step).

```
0.0 Initial Eh (TITR_EH0)
```

The starting Eh, or the Eh of a pH titration.

30.0 Eh interval (in mV) (TITR_EHD)

The increment of Eh titration (size of step).

15 Number of titration points (TITR_STEPS)

Number of titration points in step 4, Monte Carlo simulation.

3.3.3. Customizing step 1, preformatting pdb file

These entries are specifically for step 1 and do not have effect on other steps.

t Label terminal residues? (TERMINALS)

The program will identify N and C terminal residues based on the geometry if the flag is "t". When some residues are missing in the input pdb file, false terminal residues would be labeled at the "breakage". Sometimes one needs to disable this feature to work with a stand alone amino acid.

0.1 cut off water if % SAS exceeds this number (H2O_SASCUTOFF)

As surface water and salt can be taken into account by the continuum model, these molecules are recursively stripped off if they are exposed with the SAS exceeding this cutoff threshold.

2.0 distance limit of reporting clashes (CLASH_DISTANCE)

The mcce program reports atom pairs with abnormally short atom-atom distances. This line specifies the distance threshold in angstroms. Distance 2.0 is good to find atoms within bond distance and switching 2.0 to 2.4 will report possible ligand sites.

3.3.4. Customizing step 2, rotamer making

Step 2 is rotamer making, These lines are for the rules of making rotamers.

f Use control file "head1.lst" for rotamer making (ROT_SPECIF)

This line is the flag of using head1.lst or not. "t" means to use. The initial rotamer placing is controlled by either the following lines or an input file "head1.lst" created by step 1 based on the same rules and an addition file "list_rot.gold" which defines hot spot of the protein. The file "head1.lst" contains residue specific rotamer making rules. One can edit this file, and resume the program from step 2 with the flag "t" to get more initial rotamers certain some residues.

t Do swap (stereo isotope) (ROT_SWAP)

This is the flag to make stereo-isotope of Threonine residues, in which, the OG1 and CG2 are swapped.

t Do rotate? (PACK)

When the mcce program optimizes side chain rotamers, the initial rotamers is rotated from the native conformer. This flag turns on the "rotate" subroutine to rotate the single bonds by given number of steps.

6 number of rotamers in a bond rotation (ROTATIONS)

The number of steps of a bond rotation. This entry and \$(PACK) defined above is used by the same "rotate" subroutine.

f Do swing? (SWING)

Another way of rotating single bonds is to rotate a predefined angle rather than given number of steps to complete a cycle. "swing" subroutine will do this type of rotamers if the flag is "t". Each single bond will yield two new rotamers which are clock-wise and counter-clock-wise rotamers. The rotamers made here is based on the rotamers given by "rotate" subroutine. Turning on both \$(PACK) and \$(SWING) is not recommended because the combination of two subroutines creates too many rotamers and the computation is both CPU and memory demanding.

10.0 phi in degrees of swing (PHI_SWING)

This is the rotation angle of each "swing". This entry and \$(SWING) defined above is used by the same "swing" subroutine.

1.0 SAS Threshold of deleting rotamers (SAS_CUTOFF)

Surface residues don't have rotamers because their electrostatic interactions are largely screened by solvent, and rotamers don't help the calculation very much. The residues whose fractional SAS are over the number defined in this line won't get rotamers. To use a smaller number like "0.3" to delete surface side chain rotamers.

10.0 Cutoff of self vdw in kcal/mol (VDW_CUTOFF)

After initial rotamers are placed, those with big self VDW potential (VDW within side chain and VDW with backbone) will be deleted. This number is the threshold of such deleting action. First the minimum VDW will be found in side chain conformers of a residue, and then conformers with high self energy are deleted. Only heavy atoms (non Hydrogen) are considered at this stage.

5000 number of repacks (REPACKS)

While self-energy pruning removes most inaccessible side chain conformers, "repacking" finds which conformers are more accessible than others in context of other residues. Starting from random selections of microstates (one conformer from each residue), the protein is repacked with optimizing residues sequentially. Here is the number of starting microstates.

0.03 occupancy cutoff of repacks (REPACK_CUTOFF)

If fractional counts of a conformer in the "repacking" is less than this number, the conformer gets removed.

t h-bond directed rotamer making (HDIRECTED)

This is flag of making Hydrogen bond directed rotamers. With rotamers obtained after self energy pruning, the conformers are checked with the conformers of neighboring residues. When O-O and O-N distances fall into 2.5 to 3.5 Angstroms, the pair of conformers is optimized to have O-O or O-N distance closer to 2.9 Angstroms.

36 Limit number of the H bond conformers (HDIRLIMT)

When the number of hydrogen bond directed rotamers are over this number, "better" rotamers with more ideal hydrogen bond distance are elected.

0.2 threshold for two conformers being different (HDIRDIFF)

At the end of the rotamer making, the RMSD of atoms between two side chain rotamers should exceed this number, otherwise one of them will be deleted. This entry is different from the following \$(PRUNE_RMSD) entry. Any two conformers are within the distance defined by this line is treated as similar conformers and the second one would be deleted, while two conformers have to satisfy three threshold values \$(PRUNE_RMSD), \$(PRUNE_ELE) and \$(PRUNE_VDW) to be similar conformers.

0.5 Pruning cutoff of RMSD (PRUNE_RMSD)

The conformers are pruned by the geometry and pairwise interaction vector at the end of the step 2. \$(PRUNE_RMSD) defines the geometry difference. If the maximum of the atom distance between two conformers is above this number, these two conformers are treated as non-similar conformers. When two conformers are similar conformers as defined by \$(PRUNE_RMSD), \$(PRUNE_ELE) and \$(PRUNE_VDW), the second conformer would be deleted.

1.0 Pruning cutoff of electrostatic pairwise (PRUNE_ELE)

The conformers are pruned by the geometry and pairwise interaction vector at the end of the step 2. \$(PRUNE_ELE) defines the electrostatic vector difference. If any dimension of the electrostatic pairwise interaction vector is different by this number, these two are treated as non-similar conformers. When two conformers are similar conformers as defined by \$(PRUNE_RMSD), \$(PRUNE_ELE) and \$(PRUNE_VDW), the second conformer would be deleted.

8.0 Pruning cutoff of vdw pairwise (PRUNE_VDW)

The conformers are pruned by the geometry and pairwise interaction vector at the end of the step 2. \$(PRUNE_VDW) defines the electrostatic vector difference. If any dimension of the van der Waals potential pairwise interaction vector is different by this number, these two are treated as non-similar conformers. When two conformers are similar conformers as defined by \$(PRUNE_RMSD), \$(PRUNE_ELE) and \$(PRUNE_VDW), the second conformer would be deleted.

f Do relaxation on water (RELAX_WAT)

Flag of doing water relaxation. Water molecules are given both translational and rotational degrees of freedom in this relaxation process.

3.2 Distance between water and heavy atom to move water (WATER_RELAX_THR)

If the distance between water (O) and heavy atom is bigger than this number, then move the water away.

3.3.5. Customizing step 3, energy lookup table calculation

Step 3 is using DelPhi to compute reaction field energy and electrostatic interaction. This section is set up of DelPhi program.

f Use SAS + Coulomb's law for ele interaction (QUICK_ENERGIES)

Step 3 can optionally do the energy lookup table with empirical analytical equations rather than solving PB equation. Using these equations would be very fast (a few minutes vs. a few hours by DelPhi) but the error of calculated pKa would be as large as 3 pH units. If you just want find interesting spots of a big protein or refine rotamers, then try this. To turn the empirical calculation on, mark the flag as "t".

/usr/local/mcce-1.0/bin/delphi DelPhi executable (DELPHI_EXE)

The location of the DelPhi executable. This line is configured at the time of installation. The DelPhi program in this location is modified to read only unformatted pdb file.

80.0 Solvent dielectric constant for DelPhi (EPSILON_SOLV)

The dielectric constant of solvent. Although it is editable here, modifying this entry would void the calculation as all other parameters were calibrated at dielectric constant 80.

65 Grids in each DelPhi (GRIDS_DELPHI)

The DelPhi grid size. It must be an odd number. More grid points produce more accurate result but costs much more CPU time. Since mcce program uses focusing technique, a larger number is not necessary.

2.0 The target grids per angstrom for DelPhi (GRIDS_PER_ANG)

The DelPhi focusing run will end up with this targeted resolution. A minimum value of 1.0 grid per angstrom is recommended.

1.4 Radius of the probe (RADIUS_PROBE)

The size of the probe to evaluate solvent accessible surface (SAS) and defines dielectric boundary. If this value is set to be 0, then the surface detected becomes van der Waals surface.

2.0 Ion radius (IONRAD)

Ion radius is a parameter of Poisson-Boltzmann equation solver, DelPhi.

0.15 Salt (SALT)

Salt concentration of the solvent in unit M (Moles/Liter).

f Reassign charge and radii before delphi (REASSIGN)

The mcce program will reassign charges and radii to atoms from the parameter files if this flag is "t". Usually, the charges and radii are read in from extended pdb file "step2_out.pdb", in which these parameters are already assigned but editable.

f Recalculate torsion energy when write out head3.lst (RECALC_TORS)

If this flag is "t", then recalculate torsion energy for all conformers even when step 3 only calculates a part of conformers.

3.3.6. Customizing step 4, Monte Carlo sampling

In this step, the mcce program first loads self energy of conformers from file "head3.lst" and pairwise interaction from directory "energies", then simulate the titration with Monte Carlo sampling.

```
f          Average the pairwise, "f" uses the smaller          (AVERAGE_PAIRWISE)
```

The electrostatic pairwise interaction calculated in two directions are slightly different. We can either average them or choose the smaller (absolute) value. As the difference may be from the inaccurate dielectric boundary, using the smaller value reduces the dielectric boundary error.

```
20.       Warning Threshold of difference in pairwise          (WARN_PAIRWISE)
```

When the difference of pairwise interaction in two directions is very different, that is, absolute value bigger than 0.5 kcal/mol and percentage difference is bigger than this number, the program will issue warning message.

5.0 Big pairwise threshold to make big list (BIG_PAIRWISE)

Multiple-flips happen on the big list of a residue, which enables proton transfer from site to site, instead of site to solvent. The big list is the neighbor list with big pairwise interaction. The threshold here defined here is used to make such big lists for each residue.

```
-1        Random seed, -1 uses time as random seed            (MONTE_SEED)
```

Random seed, -1 to use time as the random seed.

```
298.15    Temperature                                          (MONTE_T)
```

The temperature of Monte Carlo simulation. As the reference standard free energy (pKa of acid in solution) is at room temperature, using other temperatures may require modifications on other parameters.

```
3         Number of flips                                      (MONTE_FLIPS)
```

A new microstate can be created by flipping conformers of one or multiple residues. The strategy we used is to flip one residue and its neighbors so that concerted motion can be properly sampled. The number here is the maximum of flipping sites.

```
100       Annealing = n_start * confs                          (MONTE_NSTART)
```

The steps of annealing. An independent Monte Carlo sampling goes through annealing, reducing, and equilibration stages. The sampling starts from a totally random microstate which might be at "high" energy state, and gradually "cooled" down to equilibrated states. The microstates in the annealing stage would be discarded.

```
1000      Equalibration = n_eq * confs                         (MONTE_NEQ)
```

The length (steps) of reducing stage. After reducing stage, those conformers who are not occupied or 100% occupied will be given such occupancy and not be sampled.

```
0.001     Cut-off occupancy of the reduction                  (MONTE_REDUCE)
```

Some conformers are never selected in the reducing stage, so they will be flagged and not be sampled any more. The cutoff decision is based on the value defined here, a fractional number of the occupancy. The total occupancy of conformers in a residue is 1 unless all of them are fixed.

The following 4 lines are for basic Monte Carlo sampling.

```
6         Number of independent monte carlo sampling          (MONTE_RUNS)
```

The number of independent Monte Carlo samplings at one titration point. The standard deviation of these independent runs will be reported for each conformer in progress file "mc_out" and the biggest standard deviation will be output to stdout.

```
5000      Sampling = n_iter * confs                           (MONTE_NITER)
```

The number of steps in sampling at the equilibration stage. The conformer occupancy statistics is collected at this stage.

5000 Trace energy each MONTE_TRACE steps, 0 no trace (MONTE_TRACE)

The interval of tracing the running microstate energy. Every this number of steps, the mcce program will write out the microstate energy to file "mc_out". You can see how this energy fluctuates and decide if the sampling is trapped in local minima. In addition, the average of this energy is the system energy corresponding to Enthalpy H.

1000000 Maximum microstates for analytical solution (NSTATE_MAX)

If the microstates are less than this number, calculate the conformer occupancy analytically, otherwise, use Monte Carlo sampling..

The following lines are for advanced Monte Carlo sampling. The advanced Monte Carlo sampling uses less time by detecting early convergence.

f Using Yifan's monte carlo (MONTE_ADV_OPT)

The flag to turn on advanced Monte Carlo sampling.

f Using format from old version (MONTE_OLD_INPUT)

Flag it with "t" to load energy terms from old MCCE program.

5000 Min Sampling = n_iter * confs (MONTE_NITER_MIN)

The minimal number of steps of Monte Carlo sampling. No termination before this number of steps.

-1 Max Sampling = n_iter * confs(-1 stop @ converged) (MONTE_NITER_MAX)

The maximal number of steps of Monte Carlo sampling. -1 means indefinite sampling until converged.

10000 Number of iterations each cycle (MONTE_NITER_CYCLE)

One cycle is a set of Monte Carlo runs without book keeping, convergence checking etc., in order to save time.

1000 niter * n_conf check convergence (MONTE_NITER_CHK)

Convergence is checked every x cycle. $x = (\text{int})(\text{this number} * (\text{number of conformer}) / \$(\text{MONTE_NITER_CYCLE})) + 1$

-1 Number of the reduce steps(-1 stop @ converge) (MONTE_N_REDUCE)

A reduce step is a separate MC run restarting from a different random microstate. Conformers with occupancy < $\$(\text{MONTE_REDUCE})$ is fixed to have occupancy 0, therefore no longer sampled. Convergence will be checked again between different reduce runs. New reduce run is initialized until number of reduce run reaches this number ($\$(\text{MONTE_N_REDUCE}) > 0$) or convergence criteria is satisfied ($\$(\text{MONTE_N_REDUCE}) < 0$).

0.01 Threshold for convergence (MONTE_CONVERGE)

This is convergence criteria both for single monte carlo run (explained in $\$(\text{MONTE_NITER_MAX})$) and for reduce steps (explained in $\$(\text{MONTE_N_REDUCE})$). RMSD is get from comparing average occupancy at this checking point to average occupancy at last checking point. When RMSD of all active conformer occupancy is smaller than this number, convergence criteria is satisfied.

f Calculate free energy (MONTE_DO_ENERGY)

If this flag is "t", free energy subroutine is called. Caution, free energy calculation is expensive.

Free energy is calculated by non-linear average of energies from all the unique microstates sampled in MC. Therefore unique microstates are collected. And in every MC step the occupied microstate is checked against all collected microstates to see if it is a new unique one.

298.15 Starting temperature for annealing (ANNEAL_TEMP_START)

Annealing can be run in a different temperature (usually high) before data collection to bring the system out of local minimum. This parameter gives starting temperature of annealing step. Final temperature is $\$(\text{MONTE_T})$.

In step 2, optimization was done with Coulomb's law for the electrostatic interactions. This value is the dielectric constant for Coulomb's law for this purpose.

3.4. Tricks of using "run.prm"

3.4.1. Detecting ligand

As the ligand bond distance is around 1.8 to 2.4 angstroms, we can take advantage of the clash report in step 1 of the mcce program. If \$(CLASH_DISTANCE) is set to be 2.4, then run only step 1. The atom pairs within this distance will be reported.

3.4.2. Updating head3.lst

If you want to use different parameters such as reference pKa and reference reaction field energy in step 4 Monte Carlo sampling, you may edit step 3 output file "head3.lst". Another way to load new parameters and update "head3.lst" is to run step 3 without doing "Delphi". In another situation, if conformers are split into multiple directories to run the step 3, you need to "merge" them and update "head3.lst". To do this, make \$(DELPHI_END) a smaller number than \$(DELPHI_START) and run step 3 only. The self energy will be reloaded and updated based on the current parameters.

3.5. Making parameters for a new cofactor

A parameter file (tpl file) contains information about the cofactor including connectivity, physical properties such as atom sizes and charges, chemical properties such as the pKa values, and rules for making rotamers. The file name must have extension ".tpl" to be effective.

Here are three major steps of making a complete tpl file.

3.5.1. Make a basic tpl file

First of all, in order to make a tpl file for a new group (residue or cofactor), you need to extract the coordinate lines corresponding to this group in a pdb file, and make it a new pdb file. You should prepare a working directory to run mcce program for this new pdb file. For more details, you can do that as following: (For example, Let's make a tpl file "no3.tpl")

1. Copy run.prm file and the new pdb file to your working directory.

```
~~~~~ Here is the NO3.pdb including a NO3 cofactor, grepped from 1A4C.pdb ~~~~~
```

```
HETATM 3620  N   NO3  C   3       20.959  -2.193   7.806   1.00  32.48           N
HETATM 3621  O1  NO3  C   3       20.007  -1.433   7.784   1.00  27.51           O
HETATM 3622  O2  NO3  C   3       21.063  -2.989   6.897   1.00  36.49           O
HETATM 3623  O3  NO3  C   3       21.553  -2.395   8.858   1.00  29.45           O
~~~~~
```

2. Edit run.prm:

- Put the new pdb name, in this example NO3.pdb, to the line of "(INPPDB)".
- Set the flag of step 1 to be "t", other steps to be "f".
- If you want to combine several residues into one cofactor (heme and its ligands) or split one group into separate ionizable residues (propionates in a heme), copy the "name.txt" file from mcce directory to the working directory, and then point the path in line (RENAME_RULES) to this file. In this "name.txt" file, you can manipulate the names and sequence numbers of residues in the following example.

```
~~~ Here is an example of file "name.txt": ~~~
```

```
...
####_###_#_###  ####_###_#_###
*D*****        *H*****          rename all atom D to H
**** CU A 005   ****CUB*A*005
****HIS A 284  a****CUB*A*005
****HIS A 333  b****CUB*A*005
****HIS A 334  c****CUB*A*005
```

```
1****HIS A 284 1****CUB*A*005
2****HIS A 284 2****CUB*A*005
```

~~~~~

In this example, CU and several HIS residues are a part the same cofactor called CUB.

3. Run mcce program: It will take a short time to get results from step1\_out.pdb. Confirm all atoms are converted and loaded into step1\_out.pdb. If there is no parameter for the new cofactor in the mcce parameter directory, you will see a new file named "new.tpl" in your working directory. This file contains the guessed connectivity based on inter-atom distances:

~~~ This is the output file "new.tpl" ~~~~

```
### This is a temporary parameter file made for residue NO3 ###
### Make sure that all the parameters are verified before using this file as a global parameter file ###
CONFLIST NO3          NO3BK

NATOM      NO3BK      4

IATOM      NO3BK  N      0
IATOM      NO3BK  O1     1
IATOM      NO3BK  O2     2
IATOM      NO3BK  O3     3

ATOMNAME   NO3BK      0  N
ATOMNAME   NO3BK      1  O1
ATOMNAME   NO3BK      2  O2
ATOMNAME   NO3BK      3  O3

CONNECT    NO3BK  N   ion      0   O1   0   O2   0   O3
CONNECT    NO3BK  O1  ion      0   N    0   O2   0   O3
CONNECT    NO3BK  O2  ion      0   N    0   O1   0   O3
CONNECT    NO3BK  O3  ion      0   N    0   O1   0   O2
```

~~~~~

The most basic section in a parameter file is the connectivity. However, the connectivity guessed by the program and given in "new.tpl" lacks orbital type and Hydrogen. Sometimes bonds are not correctly identified. You should assign the bond orbital types (s, sp, sp<sup>2</sup>, sp<sup>3</sup> and d<sup>2</sup>sp<sup>3</sup> are currently supported), correct the falsely bonded atoms if any, and add hydrogen atoms.

As most pdb files don't have hydrogen atoms, you need to complete them in the connectivity table. To do this, we need to know where these hydrogens are and give them the right names. Some third party programs like Gaussian View can show you hydrogen positions. To name the hydrogen names right, we will follow the pdb rule ([http://www.rcsb.org/pdb/info.html#File\\_Formats\\_and\\_Standards](http://www.rcsb.org/pdb/info.html#File_Formats_and_Standards)). Usually the hydrogen atoms are named based on their connected non-hydrogen atoms. Programs like Babel (<http://mercury.aichem.arizona.edu/babel.html>) can convert from the Gaussian atom names to PDB atom names.

For each atom, there are two fields for every atom it is connected to. For the line of atom O3, in the example above, the connected N is defined by two fields "0" and "N". "0" means the connected atom is inside the same group. "N" means the connected atom name is "N". If an atom is connected to an atom from another residue or cofactor, we need account for this. The first field could should be "LIG", and the second field should be the atom name if it known or a question mark if the atom is not known.

The cofactor may have more than one ionization states. Furthermore, one ionization state can have several isomers with different connectivity (for example the proton could be on two possible positions for the netural histidine). The above situations give rise to different conformer types for one factor. Each conformer type needs its own unique 2-character ID, which should follow the cofactor name in the connectivity line. The convention is the first character is the sign of the netcharge (0, -, +), and the second one is some unique identification character.

You can get help from the link "<http://xray.bmc.uu.se/hicup/>" when you check the connectivity.

4. Make ionization states and isomers: When a bond is broken in a residue (such as protonation/deprotonation, HIS isomers), we need to prepare different connectivity tables for them. They are called different conformer types. Here is an example that NO<sub>3</sub> may be ionized as NO<sub>3</sub><sup>-</sup>, and protonated as HNO<sub>3</sub>. The tpl file is then modified as following;

~~~~~ This is "no3.tpl" file after making two conformer types of NO3 ~~~~~

```
CONFLIST NO3          NO3BK

NATOM   NO3BK        4

IATOM   NO3BK  N      0
IATOM   NO3BK  O1     1
IATOM   NO3BK  O2     2
IATOM   NO3BK  O3     3

ATOMNAME NO3BK      0  N
ATOMNAME NO3BK      1  O1
ATOMNAME NO3BK      2  O2
ATOMNAME NO3BK      3  O3

CONNECT NO3-1  N  sp2      0  O1  0  O2  0  O3
CONNECT NO3-1  O1 sp3      0  N
CONNECT NO3-1  O2 sp3      0  N
CONNECT NO3-1  O3 sp3      0  N
CONNECT NO301  N  sp2      0  O1  0  O2  0  O3
CONNECT NO301  O1 sp3      0  N  0  H1
CONNECT NO301  O2 sp3      0  N
CONNECT NO301  O3 sp3      0  N
CONNECT NO301  H1 s        0  O1
```

~~~~~

5. After editing CONNECT entries, you need to update these four sections: CONFLIST, NATOM, IATOM and ATOMNAME. It can be automatically done by running program "mk\_iatom". The way to run is "mk\_iatom new.tpl outfile.tpl", and the result in outfile.tpl looks like:

~~~~~ This is "no3.tpl" file after mk\_iatom new.tpl no3.tpl ~~~~~

```
CONFLIST NO3          NO3BK NO3-1 NO301

NATOM   NO3BK        0
NATOM   NO3-1        4
NATOM   NO301        5

IATOM   NO3-1  N      0
IATOM   NO3-1  O1     1
IATOM   NO3-1  O2     2
IATOM   NO3-1  O3     3
IATOM   NO301  N      0
IATOM   NO301  O1     1
IATOM   NO301  O2     2
IATOM   NO301  O3     3
IATOM   NO301  H1     4

ATOMNAME NO3-1      0  N
ATOMNAME NO3-1      1  O1
ATOMNAME NO3-1      2  O2
ATOMNAME NO3-1      3  O3
ATOMNAME NO301      0  N
ATOMNAME NO301      1  O1
ATOMNAME NO301      2  O2
ATOMNAME NO301      3  O3
ATOMNAME NO301      4  H1
```

```
### This is a temporary parameter file made for residue NO3 ###
### Make sure that all the parameters are verified before using this file as a global parameter file ###
CONNECT NO3-1  N  sp2      0  O1  0  O2  0  O3
CONNECT NO3-1  O1 sp3      0  N
CONNECT NO3-1  O2 sp3      0  N
CONNECT NO3-1  O3 sp3      0  N
CONNECT NO301  N  sp2      0  O1  0  O2  0  O3
CONNECT NO301  O1 sp3      0  N  0  H1
CONNECT NO301  O2 sp3      0  N
CONNECT NO301  O3 sp3      0  N
```

CONFLIST is a list of conformer types, with the first one as backbone conformer and the rest as side chain conformers. The backbone conformer is a special conformer containing atoms that do not vary in position or charge. It should be listed in CONFLIST even though no atoms are really assigned as backbone atoms. The other conformer types are obtained from CONNECT section.

NATOM is the number of atoms for each conformer type.

IATOM is the index for searching atom slot from known atom name.

ATOMNAME is the backward searching index for atom name from atom slot number.

The program requires both IATOM and ATOMNAME to efficiently search atoms and fill in missing atoms.

6 VERIFY tpl file at anytime, HOW TO LOAD THE TPL FILE AND TEST

6. Now we need to put the basic physical and chemical properties of this cofactor into tpl file. These include the number of protons transferred relative to the ground state (this can be defined arbitrarily, but the ground state is conventionally one of the neutral conformers), solution pKa, the number of electrons transferred relative to the ground state (this can be defined arbitrarily, but the ground state is conventionally one of the neutral conformers), Solution Em value, reference reaction field energy (the reaction field energy of every conformer type of the cofactor alone in solution). The format of these entries can be seen in the following example or copied from an existing tpl file from mcce parameter directory. You also need to put the atomic charges and radii into the tpl file.

~~~~~ This is the complete no3. tpl ~~~~~

```
CONFLIST NO3          NO3BK NO3-1 NO301
```

```
NATOM   NO3BK        0
NATOM   NO3-1        4
NATOM   NO301        4
```

```
IATOM   NO3-1  N    0
IATOM   NO3-1  O1   1
IATOM   NO3-1  O2   2
IATOM   NO3-1  O3   3
IATOM   NO301  N    0
IATOM   NO301  O1   1
IATOM   NO301  O2   2
IATOM   NO301  O3   3
```

```
ATOMNAME NO3-1      0  N
ATOMNAME NO3-1      1  O1
ATOMNAME NO3-1      2  O2
ATOMNAME NO3-1      3  O3
ATOMNAME NO301      0  N
ATOMNAME NO301      1  O1
ATOMNAME NO301      2  O2
ATOMNAME NO301      3  O3
```

### This is a temporary parameter file made for residue NO3 ###

### Make sure that all the parameters are verified before using this file as a global parameter file ###

#Basic Conformer Information: name, pka, em, rxn.

#23456789A123456789B123456789C

```
PROTON   NO301        0
PKA      NO301        0.0
ELECTRON NO301        0
EM       NO301        0.0
RXN      NO301        0.0
```

```
PROTON   NO3-1       -1
PKA      NO3-1        0.0
ELECTRON NO3-1        1
EM       NO3-1        0.0
```

```

RXN      NO3-1      0.0

CONNECT NO3-1  N   sp2      0   O1   0   O2   0   O3
CONNECT NO3-1  O1  sp3      0   N
CONNECT NO3-1  O2  sp3      0   N
CONNECT NO3-1  O3  sp3      0   N
CONNECT NO301 N   sp2      0   O1   0   O2   0   O3
CONNECT NO301 O1  sp3      0   N
CONNECT NO301 O2  sp3      0   N
CONNECT NO301 O3  sp3      0   N

```

```

#3.Atom Parameters: Partial Charges and Radii
# Radii from "Bondi, J.Phys.Chem., 68, 441, 1964."

```

```

RADIUS  NO3      N   1.50
RADIUS  NO3      O1  1.40
RADIUS  NO3      O2  1.40
RADIUS  NO3      O3  1.40

```

```

CHARGE  NO301   N   1.291
CHARGE  NO301   O1 -0.597
CHARGE  NO301   O2 -0.597
CHARGE  NO301   O3 -0.097
CHARGE  NO3-1   N   0.905
CHARGE  NO3-1   O1 -0.635
CHARGE  NO3-1   O2 -0.635
CHARGE  NO3-1   O3 -0.635

```

8. Put this new tpl file into param directory (your local param directory is recommended). Remove the file "new.tpl" in the current directory. Then run mcce (step1 and step2) again. If any warning message is found in the file "run.log" or "debug.log", follow the warning messages and restart mcce (step1 and step2) until no warning message is shown.

### 3.5.2. Adding atomic charges into the tpl files

Normally, if you know the structure and have a tpl file coming from Part I, you can do the following to complete the tpl file.

1. Run mcce(step 1 and step 2), make sure you can get the file step1\_out.pdb and step2\_out.pdb.
2. Transfer the file format. Use program such as babel to transfer the format of an output file with atomic charges (e.g. the output files from program gaussian) to the pdb format. We can use this command to transfer formats in babel: \$babel -i gauout abcd.log -o pdb abcd.pdb .
3. Transfer the atom names. The program match\_atoms can do most of the job. Run \$match\_atoms [file1] [file2]. (File 1 is the file from last step showing the pdb format. File 2 is step1\_out.pdb). This program will print the matching atomic informations one by one. Check the output file matched.pdb. Note: This is important to check the matching information (from the screen) with the real structure. Sometimes, if some small optimization have been done to the original pdb crystal structure to get charges in some programs, the matching information of proton in matched.pdb file may be not right. In some other cases, some atoms will not be matched, check those atom names.
3. Now you have the charge part from some other program, and the pdb-format atom names. Try to use excel to match them. Paste this information into the tpl file in the following format:

```

#3.Atom Parameters: Partial Charges and Radii
CHARGE  HEA01  aCB  -0.065
CHARGE  HEA01  1HB   0.150
CHARGE  HEA01  2HB   0.150
CHARGE  HEA01  aCG  -0.283

```

4. Now you can add the radius part also in tpl file following the following format:

```

# Radii from "Bondi, J.Phys.Chem., 68, 441, 1964."
RADIUS  HEA      FE   1.45
RADIUS  HEA      CHA  1.80
RADIUS  HEA      CHB  1.80
RADIUS  HEA      CHC  1.80

```

### 3.5.3 Get pKa, Em, charges and calculated energies.

After Part I and Part II, actually you are ready to run mcce now. First of all, you should put the new tpl file into your param directory. Then run mcce from step 1 to step 4 (modify run.prm file). The files below are given to check charges, pKa, Em and pairwise interactions :

|                   |       |                  |
|-------------------|-------|------------------|
| results           |       | files            |
| charges           | ----- | sum_crg.out      |
| pKa               | ----- | pK.out           |
| EM                | ----- | pK.out           |
| pairwise energies | ----- | ./energies/*.opp |
| occupancy         | ----- | fort.38          |
| residues surface  | ----- | acc.res          |
| atom surface      | ----- | acc.atm          |
| rotamer making    | ----- | rot_stat         |

Note:

Before running mcce program, if you want to calculate pKa or Em, you have to edit run.prm. There is a line labeling ph or Eh in step4 of file run.prm. Compare the different below:

```
...
ph      "ph" fpr pH titration, "eh" for eh titration      (TITR_TYPE)
0.0    Initial pH                                         (TITR_PH0)
1.0    pH interval                                        (TITR_PHD)
0.0    Initial Eh                                         (TITR_EH0)
30.0   Eh interval (in mV)                                (TITR_EHD)
1      Number of titration points                          (TITR_STEPS)
...
```

```
...
eh      "ph" fpr pH titration, "eh" for eh titration      (TITR_TYPE)
0.0    Initial pH                                         (TITR_PH0)
1.0    pH interval                                        (TITR_PHD)
0.0    Initial Eh                                         (TITR_EH0)
30.0   Eh interval (in mV)                                (TITR_EHD)
1      Number of titration points                          (TITR_STEPS)
...
```

## 3.6. Running MCCE on a computer cluster

## 4. Understanding MCCE results

### 4.1. MCCE Physics

### 4.2. MCCE program

### 4.3. Calculating pKa

### 4.4. Calculating Em

### 4.5. Proton uptake upon reduction

### 4.6. The controlling factors of pKa and Em

### 4.7. Free energy of ionizing a residue at specific pH and Eh

### 4.8. MCCE tools

## **5. Trouble Shooting**

### **Appendix: Format of Input and Output files**

---

*Last Updated: 09/15/2004, Gunner's Lab, City College of New York, 2004*