VDW Calculation in MCCE

How VDW is calculated and scaled

Parameters (General form)

$$\mathcal{V}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] = \varepsilon\left[\left(\frac{R_{min}}{r}\right)^{12} - 2\left(\frac{R_{min}}{r}\right)^{6}\right]$$

$$\sigma_{AB} = \frac{\sigma_{AA} + \sigma_{BB}}{2}$$

$$\varepsilon_{AB} = \sqrt{\varepsilon_{AA}\varepsilon_{BB}}$$

Parameters VDW_RAD and VDW_EPS are assigned in every atom of every conformer, and are combined to calculate VDW in step 3.

Parameter location: 00alwaysneeded.tpl

Function: vdw_sim() in vdw.c

Parameters (A,B form)

AB coefficient:

$$V_{\mathrm{LJ}}(r) = \frac{A}{r^{12}} - \frac{B}{r^{6}},$$

A=C12, B=C6

A,B are based on atom types, precalculated and efficient, but lacks flexibility.

This parameter set is located in file amber.tpl or 00always_needed.tpl.

MCCE PDB format

```
ATOM     25  CG1 ILE A0004_001  19.774  40.976  46.525   2.000       0.000      01O000M000
```

The atom record is column based.

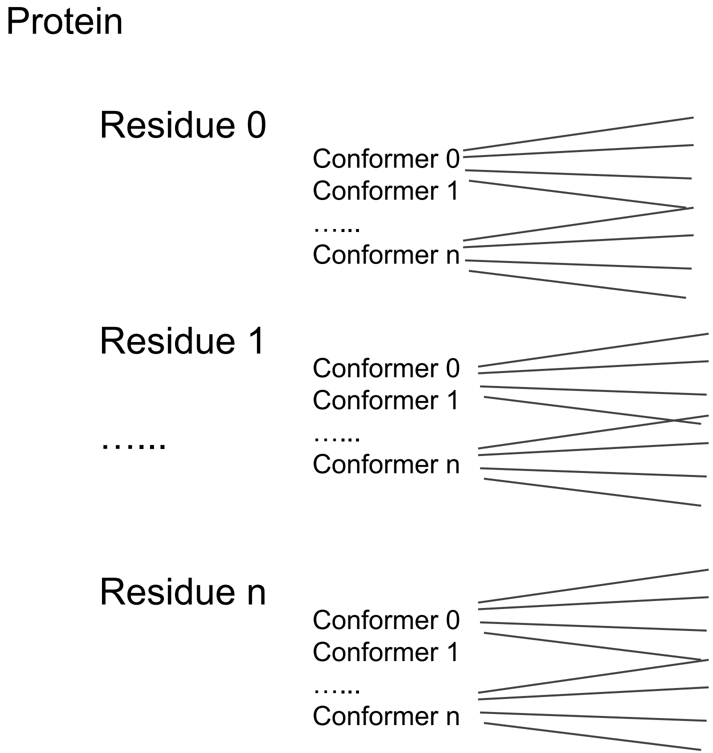
Atom name, residue name, chain ID, sequence number, insertion code, conformer number, xyz coordinates, ele boundary radius, charge, conformer history.

Conformer history:

First 2 char: conformer type, BK, 01, -1, +1 etc
3rd char: how it was made. O for original, R for rotamer etc

Note: we can not count on chain ID and sequence number alone to determine a residue for two reasons: insertion code and MCCE split residues (TERMINAL and HEM for example)

ATOM record data structure



Protein

Residue 0

Conformer 0
Conformer 1
……
Conformer n

Residue 1

Conformer 0
Conformer 1
……
Conformer n

……

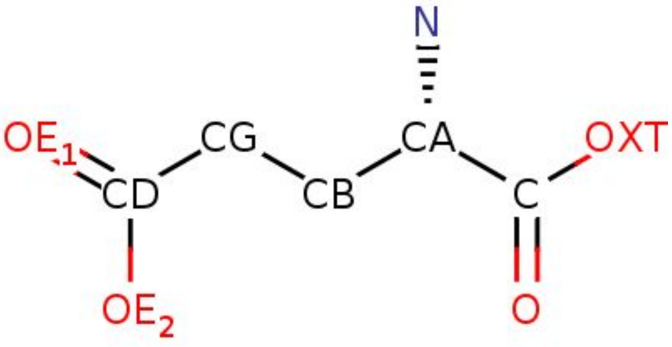
Residue n

Conformer 0
Conformer 1
……
Conformer n

Note:

1. Conformer 0 is special. It is considered as backbone.
2. Conformer 0 vdw is not recorded individually, it is treated as a whole.

1-2 Connectivity



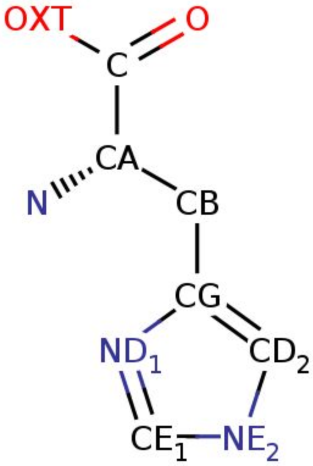
MCCE calculates most 1-2 connectivity based on the tpl file CONNECT definition. So even when atoms are missing, MCCE still knows how atoms are supposedly connected.

For special case, such as di-sulfur bridge in CYS and ligand binding residue, MCCE uses atom distance as assistance to find bonded atom.

1-3 Connectivity

- 1-3 Connectivity is extended based on 1-2 connectivity.
- 1-2 connected and 1-3 connected atoms have vdw=0

1-4 Connectivity



- 1-4 connectivity is extended based on 1-2 connectivity.
- 1-4 connected atoms had reduced vdw. The reduction factor is defined by FACTOR_14LJ in run.prm.
- 1-2, 1-3 and 1-4 connectivity go through backbone.
- In case of atoms are in a ring, lower order connectivity is used. For example, ND1 to CD2 are 1-3 connection.

Atom to atom vdw

Cutoff near = 1, e = 999

Cutoff far = 10, e = 0

C12, C6 method are used.

Function: vdw(atom1, atom2)

Conformer to conformer vdw

1. Find connectivity
2. Sum up atom to atom vdw
3. If conformers are the same one, reduce to ½ as it was a over calculated in this case. Why is it overcalculated?

Function: vdw_conf(conf1, conf2)

VDW0 - Intra conformer vdw

Conformers within a residue (except backbone conformer) can only exist one at a time, they do not interact and therefore vdw is 0.

Atoms within a conformer can have vdw. Which way is better? Reduce interaction inside or outside vdw_conf()?

VDW1 to backbone

Conformer 0 are grouped as one. So even if a residue does not have a backbone, we still need to make a BK conformer in tpl file in order to reserve this conformer slot.

VDW1 of a conformer (conformer index >=1) is calculated as this conformer’s interaction to ALL backbone conformers (conformer index = 0).

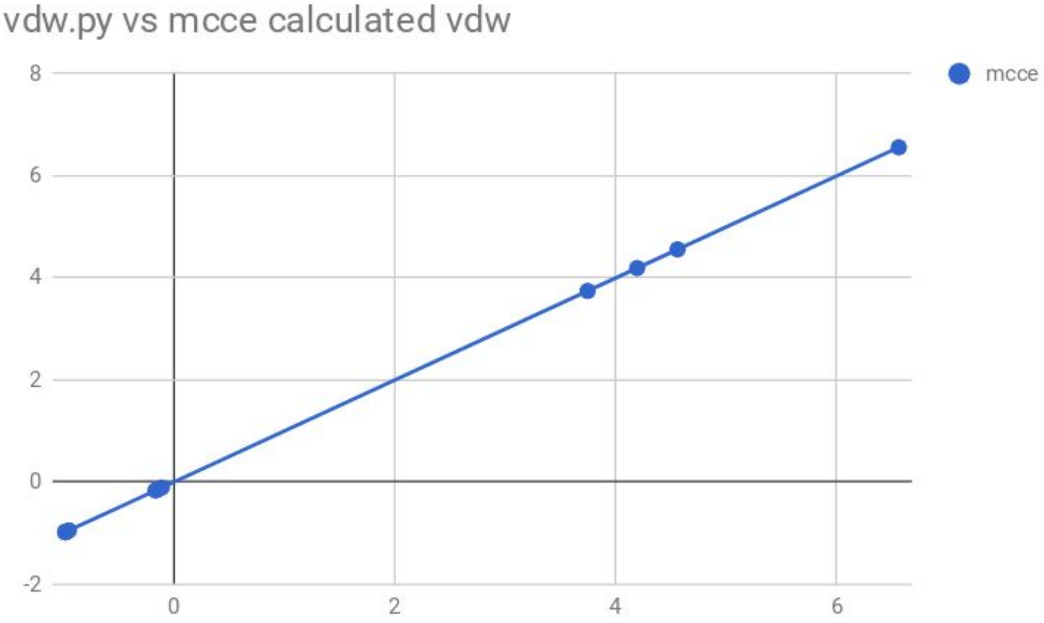
VDW between conformers

Unlike ELE pairwise interaction, VDW pairwise interaction is analytically calculated. So we only need to calculate have of the matrix and mirror to the other half.

Conformer conformer VDW is in the second pairwise interaction column of opp files.

Conformer self VDW (VDW0) and backbone VDW (VDW1) are in head3.lst

Benchmark



Scaling in step 4, Monte Carlo sampling

Scaling happens in step 4. The scaling factors are defined in extra.tpl and read in as environment variable:

```
SCALING  VDW0      1.0
SCALING  VDW1      1.0
SCALING  VDW       1.0
SCALING  TORS      1.0
```