CMMDE: An Introduction to Three New Features in ORCA 6

By CMMDE Core Team

Department of Chemistry & Biochemistry, Global Center for Science and Engineering, Waseda University, Japan
Department of Chemistry, Institut Teknologi Bandung, Indonesia
Department of Chemistry, Institut Teknologi Sumatera, Indonesia
Department of Chemistry, Universitas Pertamina, Indonesia
Masyarakat Komputasi Indonesia

August 17, 2024

Outline

Outline

CMMDE: Computational Molecular and Material Design Environment

Why did we develop CMMDE?

CMMDE was developed to reduce the difficulties of preparing input files and analyzing popular computational chemistry software outputs.

CMMDE: Computational Molecular and Material Design Environment

Why did we develop CMMDE?

CMMDE was developed to reduce the difficulties of preparing input files and analyzing popular computational chemistry software outputs.

What CMMDE can do

- Preparing the input files
- Submitting the calculations to the queueing system (if available)
- Analyzing the outputs

Outline

Implicit Solvations in Orca 6

Currently supported solvation models

- ALPB (only for XTB calculations at `GFNFF`,`GFN0XTB`, `GFN1XTB`, and `GFN1XTB` levels)
- GBSA (Also for XTB)
- CPCM
- Cosmo-based CPCM (CPCMC)
- SMD
- SMD with DRACO

Implicit Solvation in CMMDE (ammonia in water)

```
cmmde.py -i "N" -j opt -solvent water -solvtype GBSA
```

Implicit Solvation in CMMDE (Cont'd)

If we want to use other solvation models

```
cmmde.py -i "N" -j opt -solvent water -solvtype SMD
cmmde.py -i "N" -j opt -solvent water -solvtype CPCM
cmmde.py -i "N" -j opt -solvent water -solvtype CPCMC
```

If we want to activate the DRACO model

```
cmmde.py -i "N" -j opt -solvent water -solvtype SMD
-draco True
```

A Hybrid of Explicit and Implicit Solvation

Glucose solvated by 10 water molecules

```
cmmde.py -i "C(C1C(C(C(C(O1)O)O)O)O)O" -solvent water
-solvator true -nsolv 10
```

- Currently, the solvator module can only handle the ALPB solvation model.

Reoptimizing the solvation structure

```
cmmde.py -i cmmd.solvator.solventbuild.xyz -j opt -m
XTBFF -solvent water
```

- Alternative methods (`-m`): `GFN0XTB`, `GFN1XTB`, and `GFN2XTB`.

Outline

Docker: A Quantum Mechanical Docking

- The docker module can automate the binding pocket search through three levels of accuracy:
  - `SCREENING`: A low accuracy docking, good for a larger dataset for an initial screening process.
  - `NORMAL`: A moderate accuracy docking, with a moderate level of accuracy. It is good for a medium-size dataset.
  - `COMPLETE`: A robust docking with a higher level of accuracy, only for the screening of a very small dataset.

Docker: A Quantum Mechanical Docking

- The docker module can automate the binding pocket search through three levels of accuracy:
  - `SCREENING`: A low accuracy docking, good for a larger dataset for an initial screening process.
  - `NORMAL`: A moderate accuracy docking, with a moderate level of accuracy. It is good for a medium-size dataset.
  - `COMPLETE`: A robust docking with a higher level of accuracy, only for the screening of a very small dataset.

General CMMDE command for Orca6's docker

```
cmmde.py -j docker -i host.xyz -guest guest.xyz
```

Docker: A Quantum Mechanical Docking

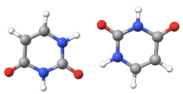
- The docker module can automate the binding pocket search through three levels of accuracy:
  - `SCREENING`: A low accuracy docking, good for a larger dataset for an initial screening process.
  - `NORMAL`: A moderate accuracy docking, with a moderate level of accuracy. It is good for a medium-size dataset.
  - `COMPLETE`: A robust docking with a higher level of accuracy, only for the screening of a very small dataset.

General CMMDE command for Orca6's docker

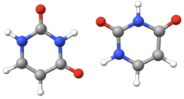
```
cmmde.py -j docker -i host.xyz -guest guest.xyz
```

For an arbitrary run, you can change the host and guest molecules above.

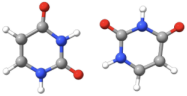
Example 1: Uracil Dimerization



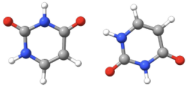
6
(0.0 kcal/mol)



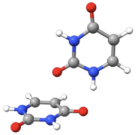
9
(2.9)



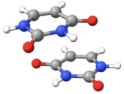
4
(4.0)



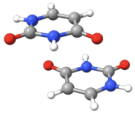
2
(5.6)



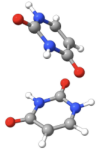
1
(6.8)



11
(8.0)



3
(8.6)



10
(11.1)

```
cmmde.py -j docker -i uracil.xyz -np 8 -guest uracil.xyz
```

Example 1: Uracil Dimerization (Cont'd)

Step 1: Geometry Optimization

- `mkdir uracil; cd uracil`
- `cmmde.py -j opt -i "C1=CNC(=O)NC1=O" -np 8`

Example 1: Uracil Dimerization (Cont'd)

Step 1: Geometry Optimization

- `mkdir uracil; cd uracil`
- `cmmde.py -j opt -i "C1=CNC(=O)NC1=O" -np 8`

Step 2: Docking

- `cd ../; mkdir UracilDimer; cd UracilDimer`
- `cmmde.py -j docker -i ../uracil/cmmd.xyz -guest ../uracil/cmmd.xyz -np 8`

- **Running time: 4 minutes 18 seconds 32 msec**
- Important message from the output: `LOWEST INTERACTION ENERGY: -18.481342 kcal/mol (structure 9)`

Example 2: Li-ion Binding Sites in Glucose

Step 1: Make a simple Li-ion Cartesian

- `mkdir li; cd li`
- `echo -e "1 \n \nLi 0 0 0" > geom.xyz`

Example 2: Li-ion Binding Sites in Glucose

Step 1: Make a simple Li-ion Cartesian

- `mkdir li; cd li`
- `echo -e "1 \n \nLi 0 0 0" > geom.xyz`

Step 2: Geometry Optimization of Glucose

- `cd ../;mkdir glucose;cd glucose`
- `cmmde.py -j opt -i "O[C@@H]1[C@H](O)[C@@H](O)[C@H](O)[C@@]([H])(CO)O1" -np 8`

Example 2: Li-ion Binding Sites in Glucose

Step 1: Make a simple Li-ion Cartesian

- `mkdir li; cd li`
- `echo -e "1 \n \nLi 0 0 0" > geom.xyz`

Step 2: Geometry Optimization of Glucose

- `cd ../;mkdir glucose;cd glucose`
- `cmmde.py -j opt -i "O[C@@H]1[C@H](O)[C@@H](O)[C@H](O)[C@@]([H])(CO)O1" -np 8`

Example 2: Li-ion Binding Sites in Glucose (Cont'd)

Step 3: Docking

- `cd ../; mkdir GlucoseLi3; cd GlucoseLi3`
- `cmmde.py -j docker -i ../glucose/cmmd.xyz -guest ../li/geom.xyz -guestcharge 1 -nrepeat 3 -np 8`

- `-nrepeat 3`: Herein, three lithium ions will be coordinated to the active sites of the glucose.
- `-guestcharge 1`: The charge of lithium-ion is $+1$.
- Important outputs:
  - `cmmd.docker.struc1.allopt.xyz`: Glucose + 1 Li^+
  - `cmmd.docker.struc2.allopt.xyz`: Glucose + 2 Li^+
  - `cmmd.docker.struc3.allopt.xyz`: Glucose + 3 Li^+
  - `cmmd.docker.xyz`: The best structure

Outline

Example 1: Cyclohexane Conformation

Step 1: Geometry optimization of cyclohexane

- `mkdir cyclohexane; cd cyclohexane`
- `cmmde.py -j opt -i "C1CCCCC1" -np 8`

Example 1: Cyclohexane Conformation

Step 1: Geometry optimization of cyclohexane

- `mkdir cyclohexane; cd cyclohexane`
- `cmmde.py -j opt -i "C1CCCCC1" -np 8`

Step 2: Global optimization of Cyclohexane

- `cd ../; mkdir GOATcyclohexane; cd GOATcyclohexane`
- `cmmde.py -j goat -i ../cyclohexane/cmmd.xyz -np 8`

Example 2: Oxidation of Glucose

Step 1: Geometry optimization of O_2

- `mkdir oxygen;cd oxygen`
- `cmmde.py -j opt -i "O=O" -mult 3 -np 8`

Example 2: Oxidation of Glucose

Step 1: Geometry optimization of O_2

- `mkdir oxygen;cd oxygen`
- `cmmde.py -j opt -i "O=O" -mult 3 -np 8`

Step 2: Docking

- `cd ../; mkdir GlucoseOxygen; cd GlucoseOxygen`
- `cmmde.py -j docker -i ../glucose/cmmd.xyz -guest ../oxygen/cmmd.xyz -m XTBFF -np 8`

Example 2: Oxidation of Glucose (Cont'd)

Step 3: Prediction of glucose oxidation products

- `cd ../; mkdir REACTGlucoseOxygen; cd REACTGlucoseOxygen`
- `cmmde.py -j GOAT-REACT -i ../GlucoseOxygen/cmmd.docker.xyz -gfn_up GFN1XTB -np 8`

Thank You!

By CMMDE Core Team

- Aditya Wibawa Sakti
- Muhamad Abdulkadir Martoprawiro
- Atthar Luqman Ivansyah
- Nova Pratiwi Indriyani
- Eduardus
- Rahmat Gunawan
- Rustaman
- Mirella Fonda Maahury
- Hasan Al-Rasyid
- Arifin
- Aulia Sukma Hutama