2014/5/21
2014年バイオスーパーコンピューティング・ソフト講習会シリーズ

「CafeMol」粗視化分子モデル計算ソフト講習会（実習編）

検崎博生
理化学研究所情報基盤センター

CafeMol (www.cafemol.org)

- Features are;
  - Various CG models
    - protein/DNA/RNA
    - multiple basin model
    - accurate CG model
  - Simulating protein-at-work
    - “switching”
- Under development
  lipid

- Developer

検崎博生、古賀信康、藤原慎司、堀直人、金田亮、李 文飛、岡崎圭一、姚 新秋、寺川剛、伊藤真志保、高田彰二
京都大学高田研究室

CafeMol 2.1 (2013/7)
source & manual released

SCLS Supercomputer system

- Supercomputational Life Science(SCLS)
  - HPCI戦略プログラム分野1
    「予測する生命科学・医療および創薬基盤」
- K computer compatible
- System
  - PRMEHPC FX10
  - 48nodes
  - CPU: SPARC64™IXfx
  - Memory: 32GB/node

How to get CafeMol(SCLS)

1. log in SCLS supercomputer system
2. copy the /home/islim/cafemol/cafemol2.1_scls directory
3. read README, README_SCLS, and INSTALL files

cafemol2.1_scls directories

src: source files

para: parameter files

aicg: aicg-related files

pdb: sample PDB files

ninfo: sample ninfo files

example: sample input/output files

How to get CafeMol(general)

1. Access “http://www.cafemol.org”
2. Download the latest version of CafeMol
3. Extract it

   $ tar zxvf CafeMol_xxx.tar.gz

Extracted directories

src: source files

para: parameter files

aicg: aicg-related files

pdb: sample PDB files

ninfo: sample ninfo files

example: sample input/output files

How to make

1. $ cd src
2. $ vi Makefile
3. $ make clean
4. $ make

Editing Makefile

- Uncomment the appropriate lines
- For SCLS and K computer, "#------ K computer" block is appropriate

FC = mpifrtpx

FC_UTIL = mpifrtpx

MPI parallelization

CPP = -DTIME –DMPI_PAR –DMPI_PAR2 –DMPI_PAR3 –DMPI_REP

INC = timing measurement

OPT = -Kfast, openmp OpenMP parallelization

LIB = optimization

How to execute CafeMol

Normal simulation

$ ./cafemol [input-file]

MPI simulation(depend on system)

$ mpirun –n [mpi-parallel-number] ./cafemol [input-file]

Restart simulation

$ ./cafemol [input-file] [restart-file]

In case of error with “segmentation fault”

$ ulimit –s unlimited

How to execute CafeMol(SCLS)

$ pjsub ./sh3.sh

sh3.sh

```
#!/bin/sh
#PJM –L “rscgrp=small”          small, large, interactive
#PJM –L “node=1”                small: 1-12nodes
#PJM –mpi “proc=1”              # of mpi parallelization
#PJM –L “elapse=1:00:00”        small: -3:00:00
#PJM –j                         merge std output and stderr output
export OMP_NUM_THREADS=1  # of OpenMP parallelization
time ./mpiexec ./cafemol example/sh3/sh3/inp
```

$ pjqstat display job status(-A:all user)

$ pjdel [JOBID] cancel job

$ pjsub --interact interactive job

e.g.

$ pjsub --interact –L “node=2” –mpi “proc=8”

$ mpiexec ./cafemol cafemol_go_replica.sh

$ exit

Example files

- Protein folding
  - sh3, sh3.sh(serial)
- Restart simulation
  - sh3_restart
- AICG2plus model
  - aicg2plus
- Multi-basin model
  - gbp_mgo1
- CG DNA model
  - dna130, dna130.sh(16 threads)
- Protein/DNA system
  - p53_DNA_flexible_local
- Replica exchange methods
  - cafemol_go_reprica, cafemol_go_replica.sh

Input file(essential block 1)

```
<<<< filenames
path = ./data                  output directory
filename = sh3                 output file names up to the suffix
OUTPUT psf pdb dcd rst         list of output file extension name
path_pdb = ./pdb               directory of reference structures
path_ini = ./pdb               directory of initial structures
path_natinfo = ./ninfo         directory of native info files
>>>>
<<<< job_cntl
i_run_mode = 2                 2:constat T, 6:REMD
i_simulate_type = 1            1:Lengevin, 2:Berendsen, 3:Nose-Hoover
i_initial_state = 1            1:random, 2:native, 3:initial, 4:BDNA, 5:CG
>>>>
<<<< unit_and_state
i_seq_read_style = 1           1:PDB, 2:sequence, 4:CG
i_go_native_read_style = 1 1:PDB, 2:native_info, 3:none
1    protein   1SRL.pdb        define unit(chain) and state(multi-basin)
>>>>
```

Input file(essential block 2)

<<<< energy_function
LOCAL(1) L_GO local energy L_GO, L_AICG2_PLUS, L_BDNA
NLOCAL(1/1) GO EXV nonlocal energy GO, EXV, AICG2, DNA, ELE
>>>>

multi-basin model
<<<< energy_function
LOCAL(1a/1a) L_GO
LOCAL(1b/1b) L_GO
NLOCAL(1a/1a) GO EXV
NLOCAL(1b/1b) GO EXV
MULTIGO_SYSTEM(1a) 1a/1a
MULTIGO_SYSTEM(1b) 1b/1b
>>>>

Input file(essential block 3)

```
<<<< md_information
n_step_sim = 1              # of switching poteintial
n_tstep(1) = 300000         # of MD steps
tstep_size = 0.2            time length in each MD step (0.05-0.2)
n_step_save = 100           frequency of output
n_step_neighbor = 100       frequency of calculate neighboring list
tempk = 300.0               temperature(K)
n_seed = 1                  random number seed(32-bit integer except 0)
>>>>
```

Input file(optional block 1)

```
<<<< initial_struct
1    1WDN_2b.pdb            initial file
>>>>
<<<< multiple_go
bdemax_mgo = 100.0          upper limit of the change in bond energy
baemax_mgo = 1.0            upper limit of the change in bond angle energy
dihemax_mgo = 0.5           upper limit of the change in dihedral angle energy
ENEGAP(1)(1)  0.0   -1.8    value of ΔV at each state
DELTA(1ab)    28.0          value of Δ
>>>>
<<<< electrostatic
cutoff_ele = 5.0            truncation distance(Debye length κD)
ionic_strength = 0.2        ionic strength(M)
diele_water = 78.0          dielectric constant(78 at 300K)
>>>>
```

CafeMol utility

- Calculation of RMSD

$./cafe calc rmsd [reference-file] [trajectory-file] [output-file] (initial-number] [last-number])

- Data conversion of restart file to text format

$ ./show_rst [restart-file] > [text-format-restart-file]

- Data conversion of text format restart file to binary file

$ ./a2rst [text-format-restart-file] [restart-file]

How to use VMD

- Linux
  - $ vmd sh3.movie
  - $ vmd sh3.psf sh3.dcd
- Windows
  - drug and drop sh3.movie to vmd icon
  - drug and drop sh3.psf and sh3.dcd to vmd icon
  - run vmd
    1. click [File]→[New Molecule…] in VMD Main
    2. click [Browse…] in Molecule File Browser
    3. choose sh3.psf and click load in Molecule File Browser
    4. click [Browse…] in Molecule File Browser
    5. choose sh3.dcd and click load in Molecule File Browser
- Representation
  1. click [Graphics]→[Representations…] in VMD Main
  2. select [Drawing Method] “VDW” in Graphical Representation
  3. change [Sphere Scale] “1.6” in Graphical Representation

実習で何をやるか

- exampleをいろいろ実行してみて、結果をVMDで観る
- exampleのinput fileを少し変更してみる
  - 計算時間を長くする：n_tstep(1)を大きくする
  - 温度を変える：tempkを低温にしたり、高温にする
  - 初期構造を変える：i_initial_state=2にすると初期構造はPDB構造
  - 乱数を変える：n_seedを0以外の整数に変える
- 適当なタンパク質のfolding simulationをする
  1. PDBからお望みのタンパク質のX線結晶構造をダウンロードして、pdbディレクトリに配置する
  2. sh3.inpやaicg2plus.inpなどを書き換え、input fileを作る
  3. ジョブスクリプトをsh3.shから書き換えジョブを投げる
- anchor, bridge, pull, boxなどのオプションを使ってみる
  - pullオプションを使って、タンパク質を引っ張ってunfoldさせる
  - boxオプションを使って、タンパク質の閉じ込め効果を調べる

Acknowledgement

CafeMol development has been supported by Research and Development of the Next-Generation Integrated Simulation of Living Matter, a part of the Development and Use of the Next-Generation Supercomputer Project of the Ministry of Education, Culture, Sports, Science and Technology.