GROMACS Tutorial

Theoretical Biophysical Chemistry Lab.

GROMACS(GROningen MAchine for Chemical Simulations) 는 네덜란드의 Groningen University에서 처음 개발되어 널리 사용되고 있는 분자 동역학 프로그램 ‘패키지’이다. GROMACS의 장점은 소스코드가 공개되어 있는 프리웨어이기 때문에 누구나 자기에게 맞는 기능을 추가해 넣을 수 있고, 조작이 비교적 쉽고, 계산이 빠르다는(특히 분자간의 non-bonded interaction에 있어서) 것이다.

1. GROMACS 설치

2013년 7월 3일 현재 GROMACS의 최신 버전은 2013년 5월 30일에 release된 4.6.2이지만 4.6.x로 버전업하면서 설치 과정이 대대적으로 개편되어 겪는 어려움 때문에 현재 필자가 Lab에서 사용하고 있는 버전은 4.5.5이다.

1-1. Configuration

사수에게 부탁하자.

1-2. Compile

Configuration을 성공적으로 수행하였다면 compile은 간단하다. GROMACS 홈페이지에서도 설명하고 있지만

./make

를 입력한 다음

./make install

을 실행해 주면 컴파일이 자동으로 완료되고 configuration에서 지정해 준 directory에 binary 및 관련 파일들이 설치된다.

2. GROMACS를 이용한 MD simulation의 기초

GROMACS가 설치된 directory의 /bin으로 들어가 설치된 binary들을 확인해 보자. 각각의 binary들은 하나의 독립된 프로그램이다. 프로그램 실행 커맨드 뒤에 -h 옵션을 줌으로써(예: grompp\_d -h, mdrun\_d -h) 각각의 프로그램들이 무슨 기능을 하는지 볼 수 있다. 기본적으로 GROMACS manual에 그대로 실려 있는 내용들로, 상당히 자세한 설명이므로 프로그램을 실행하기 전에 한 번쯤 읽어 보면 많은 도움이 될 것이다.

용매 안에서 단백질을 simulation하기 위한 flow chart와 사용되는 프로그램들이 그림 1에 나타내어져 있다. 앞에서도 설명했듯이 각각의 단계에 서로 다른 프로그램들이 사용되는 것을 볼 수 있다. 여기에 대한 세부사항은 실제로 MD simulation을 연습할 때 따로 설명하기로 한다.



그림 1. 단백질을 용매 속에서 simulation하기 위한 flow chart

(출처: GROMACS 홈페이지)

보통 MD simulation은 다음과 같은 순서로 이루어진다.

- System을 구성

- Energy Minimization

- Solvation

- Energy Minimization of solvated system

- Heating and Equilibration

- Production and Data Analysis

보통 NVT simulation(Canonical Ensemble)을 할 때는 system에 용매 효과를 더해 준 후 Temperature Coupling을 한다. 이 때 먼저 짧은 MD를 돌려 equilibration을 해 준 다음 Production을 위한 긴 MD를 돌리게 된다. Heating과 Production을 분리하는 대신 MD를 수행해서 얻어지는 분자의 움직임 데이터(trajectory)의 앞부분을 일정 분량 잘라내는 것으로 Equilibration을 대신할 수도 있다.

3. Data Structure

GROMACS에서 사용되는 파일들에는 상당히 많은 종류의 (생소한)확장자가 붙어 있기 때문에 GROMACS를 처음 접하는 사용자들에게는 혼란스러울 수도 있다. 이 절에서는 주로 사용되는 파일들의 기능을 확장자별로 설명한다. 완벽한 이해를 위해서는 force field의 구조를 비롯해 많은 세부내용들을 알아야 한다.

- pdb file

pdb file은 특정 단백질에 대한 정보들이 담겨 있는 file이다. 그 중 가장 중요한 것은 단백질을 구성하고 있는 원자들의 좌표와 amino acid residue의 배열 순서 등이다. 거의 모든 MD simulation program들이 이 file을 바탕으로 정보를 구성하며, 향후 실제 simulation을 수행할 때도 제일 먼저 pdb file을 GROMACS가 처리하기 쉬운 format으로 바꾸는 일을 할 것이다. RCSB protein data bank([www.rcsb.org](http://www.rcsb.org))에서 다양한 단백질들의 pdb file을 검색하고 내려받을 수 있다.

- gro file

GROMACS는 pdb2gmx라는 프로그램을 통해 pdb file을 gro file과 top file이라는 두 개의 독립된 file들로 분리한다. 이 중 gro file에는 system을 구성하는 원자들의 좌표가 들어 있다. 단위는 nanometer이다. gro file은 pdb file의 processing을 통해서만 나오는 것은 아니며 simulation의 결과물로 얻거나 직접 제작할 수도 있다.

- top file

top file에는 분자를 구성하고 있는 원자들의 type과 connectivity, 즉 어떤 분자들끼리 bond, angle, dihedral을 구성하고 있는지에 대한 정보가 들어 있다. GROMACS는 이를 바탕으로 force field와 system을 연결시키고 simulation을 할 수 있도록 만들어 준다.

- tpr file

tpr file은 gro file과 top file을 통해 구성한 분자의 interaction parameter들과 simulation에 사용될 여러 option들(밑의 mdp file을 참조)을 합쳐서 하나의 file로 재구성한 것이다. tpr file은 기계어(binary)이며, 그 자체로 MD simulation의 input이 된다.

- mdp file

MD simulation에 관한 여러 가지 변수들이 들어 있는 file이다. 대표적인 것들은 simulation time, time step, simulation의 종류 등이다. 변수들에 대해서는 밑에 다시 설명하기로 한다.

- cpt file

cpt는 checkpoint의 줄임말로, 중단되었거나 종료된 simulation을 끝난 지점에서 다시 시작하고 싶을 때 사용하는 file이다.

- edr, trr file

MD simulation의 결과물로 생성되는 binary file이다. edr file에는 system의 에너지와 관련된 정보, 그리고 trr file에는 system의 trajectory, 즉 원자들의 좌표와 속도에 관한 정보가 들어 있다. 일정 시간마다 simulation의 정보가 이들 file에 기록되는데, 얼마나 자주 기록할지에 대한 옵션은 mdp file에서 설정한다. 사실상 가장 중요한 정보가 담긴 file이라고 할 수 있다.

4. GROMACS로 MD simulation 해 보기

이제부터는 실제로 pdb file로부터 출발하여 MD simulation을 여러 가지 조건에서 해 보게 될 것이다. 대략적인 내용은 아래와 같다:

준비

- pdb file로부터 top file과 gro file의 생성

진공(vacuum) 중에서의 Dynamics

- Energy Minimization

- 진공에서의 MD simulation(50 ps)

- Analysis

Periodic boundary condition을 이용한 수용액상에서의 Dynamics

- Periodic Boundary Condition을 이용해 System 주위에 Rectangular Box 만들기

- 용매화된 상태에서의 Energy Minimization

- 용매화된 상태에서의 MD simulation(50 ps)

- Analysis

4-1. 준비

gmxexercise.tar 파일의 압축을 풀면 두 개의 pdb file이 보일 것이다. 우선 2ECH.pdb는 RCSB protein data bank에서 그대로 다운로드받은 파일이고, 2ECH\_mod.pdb는 그 중 model 6의 데이터만을 별도로 떼어 놓은 것이다. 보통 RCSB에서 바로 내려받은 pdb file은 이렇게 여러 가지 구조들이나 잡다한 데이터들을 포함하고 심지어 몇몇 residue들이 빠져 있는 경우도 있기 때문에 수정을 거쳐야 제대로 된 input으로 사용할 수 있다. 이제

pdb2gmx\_d -f 2ECH\_mod.pdb

를 입력하여 pdb file로부터 gro file과 top file을 만들어 보자. 이를 실행하면 GROMACS가 가지고 있는 force field들의 목록과 함께 어떤 force field를 선택할 것인지 묻는 메시지가 나온다.

1: AMBER03 protein, nucleic AMBER94 (Duan et al., J. Comp. Chem. 24, 1999-2012, 2003)

2: AMBER94 force field (Cornell et al., JACS 117, 5179-5197, 1995)

…

우선 force field로는 8번의 CHARMM27 all-atom force field (with CMAP) - version 2.0을 선택하고, water model로는 1번의 TIP3P를 선택한다. top와 gro file이 만들어지는데, 아무 옵션도 주지 않았기 때문에 file들의 이름이 미리 지정된 기본값인 topol.top와 conf.gro로 설정되었다. file들의 이름을 지정해 주고 싶을 때는 다음과 같이 입력한다.

pdb2gmx\_d -f 2ECH\_mod.pdb -o 2ECH\_mod.gro -p 2ECH\_mod.top

pdb2gmx\_d -h를 입력하면 이 옵션들에 대한 정보를 보여준다.

이제 gro file과 top file을 vi editor로 열어서 어떤 형식으로 되어 있는지 대략적으로 훑어보고 원래의 pdb file과 비교해 보자.

4-2. 진공(vacuum) 중에서의 dynamics

Energy Minimization

MD simulation으로부터 의미있는 데이터를 얻기 위해서는 본격적인 simulation을 시작하기 전에 언제나 energy minimization을 해 주어야 한다. 어떤 force field를 사용하는가에 따라서 system의 energetics가 달라지게 되고 local strain이 생겨 향후 simulation에 영향을 줄 수도 있기 때문이다. Energy minimization을 위해서는 mdrun\_d 프로그램을 짧은 시간 동안 steepest descent algorithm을 이용하여 돌린다. 우선 input file인 tpr file을 만들어 보자. tpr file은 3절에서도 설명했듯이 gro, top, mdp file을 한데 합쳐 binary로 만든 것이다. minimize.mdp 라는 file이 있는지 확인하고 vi editor로 열어 보자.

다양한 parameter들이 있는데, 중요한 것은 대강 이 정도이다.

Integrator - 수치적분 algorithm

dt - time step(단위는 ps)의 간격

steps - step의 수

nstx(v,f)out - trr 파일에 x, v, F를 몇 step마다 출력할지

emtol - energy minimization tolerance

emstep - first step for energy minimization

nstcgsteep - steepest descent frequency

tcoupl - temperature coupling(temperature를 일정하게 유지)

pcoupl - pressure coupling(pressure를 일정하게 유지)

comm\_mode - center of mass motion을 억제

epsilon\_r - dielectric coefficient of solvent

coulombtype - electrostatic interaction의 계산을 조절

vdwtype - Van der Waals interaction의 계산을 조절

grompp\_d -f minimize.mdp -c 2ECH\_mod.gro -p 2ECH\_mod.top -o 2ECH\_min.tpr

을 입력해 본다. 그러면 프로그램은 error message를 출력하는데, 이는 system이 담긴 box의 크기가 cutoff length에 비해 너무 짧아서 발생하는 것이다. Box의 크기는 gro file의 맨 마지막 줄에 지정되어 있다. gro file을 열어 이 값들을 10.0 정도로 고치고 다시 명령어를 입력하면 프로그램이 제대로 동작하게 된다. tpr file이 제대로 생성되었는지 확인해 본다.

mdrun\_d 프로그램을 통해 energy minimization을 해 보자.

mdrun\_d -s 2ECH\_min.tpr -c 2ECH\_min.gro -e 2ECH\_min.edr

약간의 시간이 지난 후 메시지와 함께 output file들이 출력된다. 출력된 gro file이 energy minimization을 거치고 난 system의 구조이다. vmd와 같은 visualization 프로그램을 이용하여 minimization을 하기 전과 후의 구조를 비교하는 것도 가능하다.

Minimization의 결과물로 gro file 외에도 edr 파일과 trr 파일이 생성되는데, 여기에는 각각 system의 energy와 trajectory(coordinate + velocity)가 저장된다. 이 file들은 binary이므로 그대로는 읽을 수 없고, gmxdump\_d라는 프로그램을 사용하여 사람이 읽을 수 있는 ASCII data로 변환해야 한다. 이런 과정을 거쳐 simulation에서 얻어진 data들을 analysis에 사용할 수 있다. 우선 edr file에 대해 간단한 분석을 해 보기로 하자.

gmxdump\_d -e 2ECH\_min.edr > energy.txt

이 명령어를 실행하면 energy.txt에 binary를 해독한 data가 저장된다. grep 명령어와 vi editor의 visual block 기능을 적절히 사용하여 x축에 시간, y축에 potential energy를 plot하여 energy minimization이 되는 것을 눈으로 확인해 보자.

이제 MD simulation을 실행할 차례이다. Energy minimization을 할 때와는 다른 parameter를 써야 한다. md.mdp를 열어 minimize.mdp와 비교해 본 후, grompp\_d를 사용하여 tpr file을 만든다.

grompp\_d -f md.mdp -c 2ECH\_min.gro -p 2ECH\_mod.top -o 2ECH\_md.tpr

주의: 분자의 connectivity는 바뀌지 않았으므로 top file은 처음 사용했던 file을 그대로 사용해도 되지만, gro file은 minimization을 통해 얻어진 것을 사용해야 한다. 이제 아까와 마찬가지로

mdrun\_d -s 2ECH\_md.tpr -c 2ECH\_md.gro -e 2ECH\_md.edr

을 입력하여 md simulation을 할 수 있다. 마찬가지로 결과물로 얻어진 edr file을 gmxdump\_d를 사용하여 변환하고 total energy를 시간에 따라 plot한다. 그래프가 이와 같이 나오는 이유는 무엇이겠는가?

지금까지의 simulation은 전부 진공 중에서 이루어졌다. 실제로 생체분자는 solvation이 되어 있는 경우가많기 때문에 대부분 수용액상에서 MD를 수행하게 된다. 이 경우 solvent effect를 고려해 주어야 한다. Solvent의 효과를 simulation에 넣기 위한 가장 간단한 방법은 dielectric screening coefficient를 설정하는 것이다. 물의 dielectric coefficient가 약 80이므로 mdp file에 epsilon\_r = 80이라는 옵션을 추가하여 다시 simulation하고, 진공 중에서의 simulation 결과와 비교해 보자.

4-3. Periodic Boundary Condition을 이용한 수용액상에서의 dynamics

MD simulation을 용액 상태에서 하기 위해서는 용매와 system과의 상호작용을 simulation에 포함시켜야 한다. 여기에는 크게 두 가지 방법이 있다. 4-2절의 마지막 부분에서처럼 dielectric한 환경으로 용매를 대신하는 implicit solvent 방법과, 실제로 물 분자를 system 주변에 둘러싼 후 simulation을 하는 explicit solvent 방법이 있다. 당연히 explicit solvent model이 실제와 더 가깝기 때문에 정확한 결과를 얻을 수 있지만, 그 대가로 계산량이 늘어나게 된다. 이 절에서는 periodic boundary condition을 이용해 system이 담긴 cubic box를 물 분자들로 채우고, 이 상태에서 MD simulation을 해 볼 것이다.

4-2절에서 얻어졌던 minimized geometry를 담은 gro file을 물 분자로 solvation시켜 보자. 이 경우 genbox\_d 라는 명령어를 사용한다.

genbox\_d -cp 2ECH\_min.gro -cs spc216.gro -o 2ECH\_solv.gro -p 2ECH.top

여기서 -cp는 용질, 즉 system을 지정해 주는 옵션이고 -cs는 용매를 지정해 주는 옵션이다. 우리는 tip3p water model을 사용하지만 spc216 water model의 gro file을 지정해 주었다. 이렇게 해도 되는 이유는 tip3p와 spc216이 같은 3-site model이기 때문에 model이 다르기 때문에 오는 차이를 짧은 시간 동안 equilibration을 통해 제거할 수 있기 때문이다. 만약 4-site model인 tip4p model이나 기타 다른 model을 사용한다면 그에 맞는 gro file을 지정해 주어야 할 것이다. 기본 제공되는 용매의 gro file들은 GROMACS가 설치된 directory의 /share/gromacs/top에 들어 있다.

이제 vmd로 생성된 2ECH\_solv.gro를 열어 보면, 정육면체형의 solvent box가 생성된 것을 확인할 수 있다. 하지만 system이 solvent box의 안에 위치하지 않고 모서리 부분에서 box 밖으로 튀어나와 있을 것이다. 이런 현상을 해결하기 위해서는 editconf\_d를 사용한다. 이 명령어는 system의 center of mass를 box 내부의 원하는 곳으로 이동시키거나, 평행 및 회전이동을 할 수 있게 해 준다. Box가 10nm 크기의 정육면체이므로 -center 5 5 5라는 옵션을 줌으로써 system의 center of mass를 box의 한가운데 갖다 놓을 수 있다.

editconf\_d -f 2ECH\_min.gro -o 2ECH\_min\_center.gro -center 5 5 5

이제 2ECH\_min\_center.gro를 다시 solvation시켜 주자. 그 전에 2ECH.top을 열어 맨 마지막 줄의 solvent 정보를 지워야 한다. 또는, 앞에서 백업된 #2ECH.top.1#을 사용해도 된다.

genbox\_d -cp 2ECH\_min\_center.gro -cs spc216.gro -o 2ECH\_solv.gro -p 2ECH.top

얻어진 2ECH\_solv.gro를 vmd로 열어 system이 box 중앙에 잘 들어갔나 확인해 보자.

이제 minimize -> md simulation을 수행할 차례이다. tpr file을 만들고 simulation을 하는 것은 앞에서와 똑같은 방법으로 할 수 있다.

grompp\_d -f minimize.mdp -c 2ECH\_solv.gro -p 2ECH\_mod.top -o 2ECH\_solv\_min.tpr

mdrun\_d -s 2ECH\_solv\_min.tpr -c 2ECH\_solv\_min.gro

grompp\_d -f md.mdp -c 2ECH\_solv\_min.gro -p 2ECH\_mod.top -o 2ECH\_solv\_md.tpr

mdrun\_d -s 2ECH\_solv\_md.tpr -c 2ECH\_solv\_md.gro -e 2ECH\_solv\_md.edr -o 2ECH\_solv\_md.trr

단순히 위 command를 copy해서 입력하는 것보다는, 한 단계에서의 output이 다음 단계의 input으로 사용되는 과정과 그 의미를 눈여겨보는 것이 좋다. 상당히 오래 걸리는 작업이므로, 작업이 진행되는 동안에 5절의 내용을 먼저 읽어보도록 하자.

md simulation에서 얻어진 trajectory인 2ECH\_solv\_md.trr을 trjconv\_d를 사용하여 gro file로 변환한 다음 vmd로 열어 보면 system이 움직이는 과정을 실시간으로 관찰할 수 있다.

5. Force Field

Force Field는 md simulation에 관한 여러 parameter의 집합체이다. 즉 bonded interaction과 non-bonded interaction을 기술하는 하나의 model이라고 할 수도 있다. 같은 system이라도 simulation에 사용된 Force Field가 다르다면 별개의 simulation이 된다. GROMACS에서는 20개 정도 되는 Force Field를 기본 제공하는데, 여기에 외부에서 가져와서 추가할 수도 있다. 시험삼아 Force Field를 추가해 보고, 이를 이용해 md simulation을 돌리려는 시도를 해 보자.

California State Univ.의 Eric J. Sorin 교수가 AMBER라는 md 프로그램의 force field를 GROMACS용으로 고친 버전이 <http://ffamber.cnsm.csulb.edu/#install> 에 업로드 되어 있다. 최신 버전을 다운받아서, GROMACS가 설치된 directory 안의 /share/gromacs/top 에 옮겨 압축을 풀고, ffamber03이라는 directory를 상위 폴더로 옮긴다.

각각의 file들에 대해 간략한 설명을 하자면

.rtp - residue database. Amino acid나 nucleic acid들에 대한 정보

.itp - interaction database. Bonded/nonbonded interaction들에 대한 정보

.hdb - hydrogen database. pdb file에 수소가 없는 경우 어떻게 수소를 추가할지에 대한 정보

.tdb - termini database. Terminal residue를 어떻게 처리할지에 대한 정보

이렇게 옮겨 주어도 GROMACS는 새로 추가된 Force Field가 있다는 사실을 인식하지 못한다. 이를 위해서는 첫째, directory의 이름이 .ff로 끝나야 하고 둘째, ffamber03 directory 내부에 forcefield.doc와 forcefield.itp라는 두 개의 file이 있어야 한다. ffamber03 directory의 이름 뒤에 .ff를 붙여 ffamber03.ff로 바꾸고, vi editor로 forcefield.doc을 새로 만들어 첫번째 줄에 Force Field의 이름을 적당히 적어 넣는다. 그리고 ffamber03.itp라는 file의 이름을 forcefield.itp로 바꿔 주면 된다.

이제 pdb2gmx\_d로 top file과 gro file을 만들어 보자. Force Field가 추가된 것을 확인할 수 있을 것이다. 하지만 이 Force Field를 선택하면 error를 출력할 것이다. Error message를 잘 읽어 보고, 아래 설명한 일련의 절차를 통해 순차적으로 error의 원인을 규명하고 수정해 주어야 한다.

- 맨 처음 나오는 error는, ffamber03 Force Field가 N-terminal과 C-terminal 처리를 제대로 하지 못하는 데서 기인한다. 사실 ffamber03은 terminal residue에 대한 정보들을 따로 가지고 있지만, pdb file에는 어떤 residue가 terminal이라고 명시되어 있지 않기 때문에 오류가 발생하게 된다. pdb file을 열어 1번 residue인 GLU와 49번 residue인 THR을 고쳐 주는데, 어떻게 고칠지는 ffamber03.ff directory 안의 ffamber03.rtp를 보고 추측해 보자.

- 이 error를 고치면, Lysine에 해당되는 LYS라는 residue가 rtp database에 존재하지 않는다는 error가 나온다. ffamber03에서 Lysine을 나타내는 residue type은 두 가지가 존재한다(LYN, LYP). 둘의 차이는 amino group의 protonation state이다. 두 residue type 중 pdb file의 상태와 일치하는 것은 무엇인가?

- 다음 error는 Histidine에 해당되는 HSE라는 residue에 대한 것이다. Histidine의 경우에도 세 가지 다른 protonation을 나타내는 3가지 residue type이 있다. 마찬가지로 pdb file을 확인하고 고쳐 주자.

- 이제 residue 4번 Serine의 HG1 atom이 rtp file과 일치하지 않는다는 error를 해결해야 한다. rtp file을 확인해 보고 pdb file의 HG1을 rtp file의 atom name에 맞게 수정해 준다.

- 마지막 error는 조금 까다로운데, rtp file에서 C-terminal인 Threonine의 부분을 직접 찾아, atom OC1을 O로, OC2를 OXT로 수정함으로써 해결할 수 있다.

새로운 force field를 사용하는 과정에서는 위에서와 같이 예상치 못한 error들이 발생할 수 있고, 원인을 찾아서 하나하나 수정해 주는 작업이 때때로 매우 번거로워질 때가 많다.