Molecular Dynamics (MD) Simulations, step by step protocol

Version 3

Parham Jabbarzadeh Kaboli,patimah Ismail,King-Hwa Ling

Abstract

To MD analysis for one of our previous results of docking, for example to analyze the drug/target, almost 40 steps are required to finalize the MD results.

The following protocol is according to our own experience in this project and was improved so that the errors were solved during the projects.

Citation: Parham Jabbarzadeh Kaboli,patimah Ismail,King-Hwa Ling Molecular Dynamics (MD) Simulations, step by step protocol. **protocols.io**

dx.doi.org/10.17504/protocols.io.mztc76n

Published: 03 Feb 2018

Guidelines

This protocol was done using GROMACS v.5.0.4 under Ubuntu Linux platform version 15.0.4, with GROMOS 53A6 force field as provided by ATB version 2.2 (Lundborg and Lindahl, 2015).

Before start

Ubuntu Linux platform version 15.0.4 must be already installed.

Then because GROMACS needs CMake and FFTW tools developed for discrete Fourier transforms (DFTs), these were installed on Ubuntu too. FFTW was configured as:

```
$ --enable-threads --enable-sse --enable-float –enable-shared
```

As installation of GROMACS is challenging and frequently got errors, it can be checked whether the GROMACS has been accurately installed by this command:

```
$ mdrun --version
```

Protocol

Preparation of topology files for target and ligand

Step 1.

After performing molecular docking, you should generate your 'docked-protein.pdb' file after which you are requiered to copy and paste the coordinates of ligand and protein into two different files.

You should start the MD work with two files, 'drug.pdb' and 'target.pdb'.

SOFTWARE PACKAGE (Windows)
VMD, 1.9.2
cmd COMMAND (Linux - Ubuntu)
`$ mdrun --version`
To Check GROMACS.

Preparation of topology files for target and ligand

Step 2.

Open drug.pdb in UCSF Chimera.

Preparation of topology files for target and ligand

Step 3.

Add H

Using top menu= Tools/Structure Editing/AddH

Preparation of topology files for target and ligand

Step 4.

Save new "drug.pdb" file. (e.g. drugH.pdb)

Preparation of topology files for target and ligand

Step 5.

Open Automated Topology Builder, and register for free.

Then login.

Preparation of topology files for target and ligand

Step 6.

Submit your "drugH.pdb".

After processing, you will find several kinds of results.

Preparation of topology files for target and ligand

Step 7.

Chose "gromos53A6" force field.

Gromacs uses United Atom Topology.

Preparation of topology files for target and ligand

Step 8.

Copy the lines of United Atoms into a topology file: **drug.itp**

Preparation of topology files for target and ligand

Step 9.

Copy the lines of United Atoms (original) into a pdb file: **drug-atb.pdb**

Preparation of topology files for target and ligand

Step 10.

Copy **drug.itp** and **drug-atb.pdb** in your project folder.

Preparation of topology files for target and ligand

Step 11.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx pdb2gmx –f target.pdb –ff gromos53a6 –ignh –ter –water spc
```

Use SPC water model for gromos force field.

Preparation of topology files for target and ligand

Step 12.

Convert "conf.gro" to "conf.pdb" with Open Babel.

Preparation of topology files for target and ligand

Step 13.

Copy the coordinates of "conf.pdb" into a new file: **complex.pdb**

Preparation of topology files for target and ligand

Step 14.

Copy the coordinates of "drug-atb.pdb" into "complex.pdb" just after the last line of protein. And save.

NOTES

Parham Jabbarzadeh Kaboli 03 Feb 2018

Sometimes it is better to have gro format. Gromacs commands work with both types of files: pdb and gro.

In case you get error, you can convert "drug-atb.pdb" into "drug-atb.gro". And again, make the "complex.gro" using "conf.gro" and "drug-atb.gro" files.

Solvation

Step 15.

COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx editconf –f complex.pdb –o newbox.gro –bt triclinic -c –d 1.0
```

NOTES

Parham Jabbarzadeh Kaboli 03 Feb 2018

In case you made "complex.gro" you should use this file in the command, instead of "complex.pdb".

Solvation

Step 16.

COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx solvate –cp newbox.gro –cs spc216.gro –p topol.top –o solv.gro
```

SAFETY INFORMATION

Using VMD, check whether your proteins covered by solvent or not. Triclinic boxes are for globular proteins. Dodecahedron can be also used if your protein is still completely inside the box and covered by solvent.

Edit Topology

Step 17.

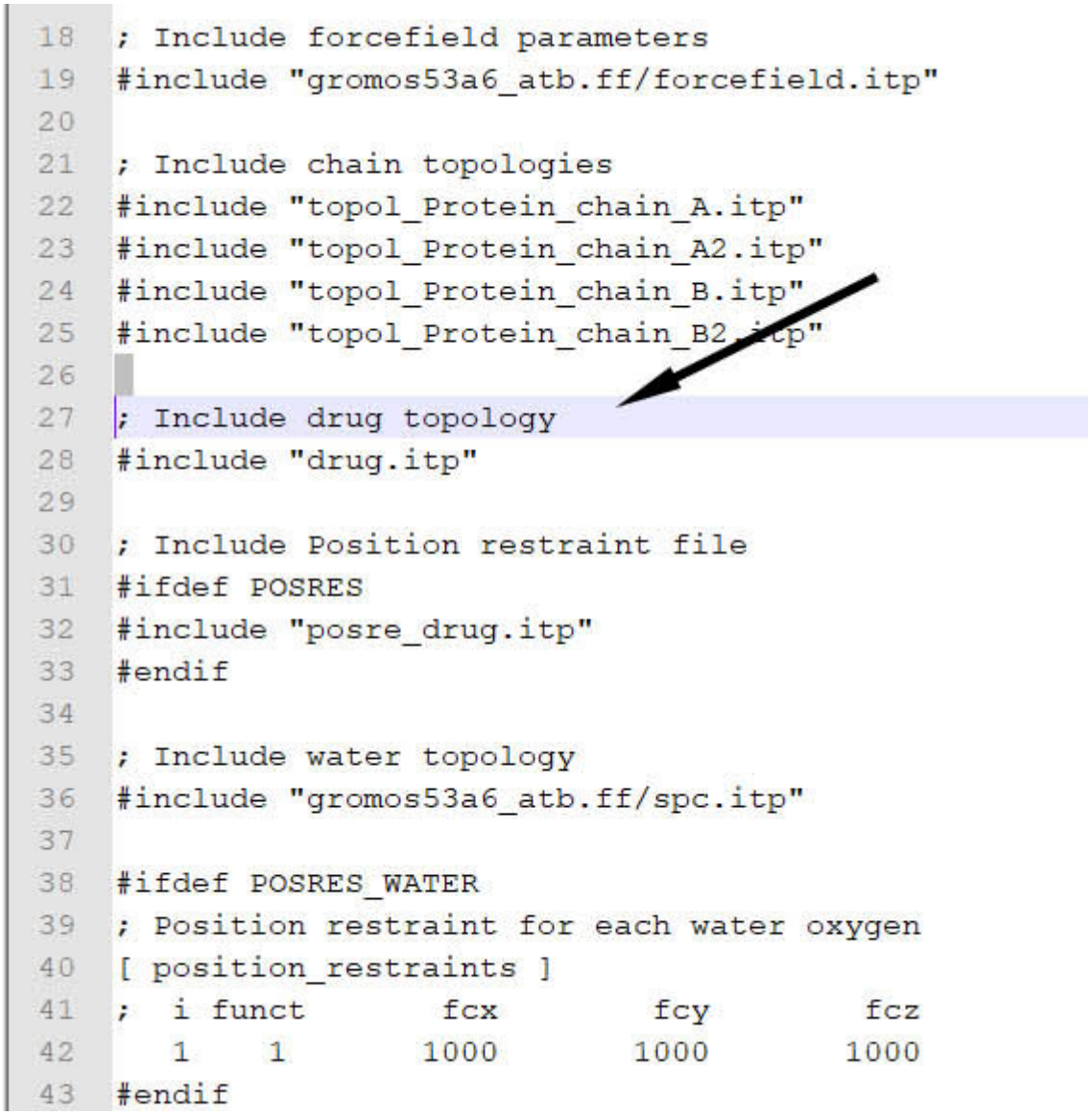
Open "topol.top".

Edit Topology

Step 18.

After ';Include chain topologies' section (in topol.top) add the following lines:

```
; Include drug topology#include 'drug.itp'
```



```
; Include forcefield parameters
#include "gromos53a6_atb.ff/forcefield.itp"

; Include chain topologies
#include "topol_Protein_chain_A.itp"
#include "topol_Protein_chain_A2.itp"
#include "topol_Protein_chain_B.itp"
#include "topol_Protein_chain_B2.itp"

; Include drug topology
#include "drug.itp"

; Include Position restraint file
#ifdef POSRES
#include "posre_drug.itp"
#endif

; Include water topology
#include "gromos53a6_atb.ff/spc.itp"

#ifdef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
;  i funct       fcx        fcy        fcz
   1    1       1000       1000       1000
#endif
```

Edit Topology

Step 19.

In the section [molecules] add a new line:

```
LIG                 1
```

```
37
38  #ifdef POSRES_WATER
39  ; Position restraint for each water oxygen
40  [ position_restraints ]
41  ;  i funct       fcx        fcy        fcz
42     1    1       1000       1000       1000
43  #endif
44
45  ; Include topology for ions
46  #include "gromos53a6_atb.ff/ions.itp"
47
48  [ system ]
49  ; Name
50  B-RAF PROTO-ONCOGENE SERINE/THREONINE-PROTEIN KINASE in water
51
52  [ molecules ]
53  ; Compound        #mols
54  Protein_chain_A     1
55  Protein_chain_A2     1
56  Protein_chain_B     1
57  Protein_chain_B2     1
58  LIG                  1
59  SOL         21507
60  CL               15
61
```

Edit Topology

Step 20.

Autosave "topol.top".

SAFETY INFORMATION

Don't change the file name of topol.top.

Saveing MD parameter files

Step 21.

Add the following lines into "**em.mdp**" and save in the project folder.

```
; $Id$; Template for energy minimisation of full system with OPLS-AA;; by
default position restraints are used (POSRES).;;
==========================================================;      DO
NOT USE THIS INPUT FILE WITHOUT CHECKING ALL SETTINGS.; *YOU* ARE SOLELY
RESPONSIBLE FOR CORRECT INPUT TO YOUR SIMULATION.;
==========================================================;; some
values from
http://code.google.com/p/acpypi/wiki/TutorialAcpypi4GromacsOPLScpp
= cppinclude        = -I. -I.. -I../topdefine         = -DFLEXIBLEintegrator
= steepemtol          = 100emstep         = 0.001nsteps         =
50000nstcgsteep     = 10;nstxout       = 100 ; Writing full precision
coordinates every nanosecond;nstvout       = 10 ; Writing velocities every
nanosecondnstlog         = 1000  ; Writing to the log file every
stepnstenergy      = 1000  ; Writing out energy information every
step;energygrps     = System        ; Which energy group(s) to write to
diskconstraints              = nonenstcomm                  = 100vdwtype
= Cut-off   ; use shift for L-BFGScoulombtype             = Reaction-
Fieldoptimize_fft             = yesns_type                  = gridrlist
= 1.4rcoulomb                 = 1.4rvdw                     = 1.4Tcoupl
= noPcoupl                   = nogen_vel                 = nopbc
= xyz
```

Saveing MD parameter files

Step 22.

Add the following lines into "**nvt.mdp**" and save in the project folder.

```
title       = Protein-ligand complex NVT equilibration define      = -DPOSRES
; position restrain the protein and ligand; Run parametersintegrator  = md
; leap-frog integratornsteps      = 150000     ; 2 * 50000 = 100 psdt
= 0.002     ; 2 fs; Output controlnstxout     = 500       ; save coordinates
every 1.0 psnstvout     = 500       ; save velocities every 1.0 psnstenergy
= 500       ; save energies every 1.0 psnstlog      = 500       ; update log
file every 1.0 psenergygrps  = Protein ; Bond parameterscontinuation    = no
; first dynamics runconstraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds)
constrainedlincs_iter      = 1             ; accuracy of LINCSlincs_order
= 4             ; also related to accuracy; Neighborsearchingcutoff-scheme
= Verletns_type         = grid      ; search neighboring grid cellsnstlist
= 10        ; 20 fs, largely irrelevant with Verletrcoulomb        = 1.4
; short-range electrostatic cutoff (in nm)rvdw            = 1.4       ;
short-range van der Waals cutoff (in nm); Electrostaticscoulombtype     = PME
; Particle Mesh Ewald for long-range electrostaticspme_order       = 4
; cubic interpolationfourierspacing  = 0.16      ; grid spacing for FFT;
Temperature couplingtcoupl      = V-rescale                    ; modified
Berendsen thermostattc-grps     = Protein_LIG Water_and_ions    ; two
```

```
coupling groups - more accuratetau_t       = 0.1   0.1                    ;
time constant, in psref_t       = 300   300                    ; reference
temperature, one for each group, in K; Pressure couplingpcoupl      = no
; no pressure coupling in NVT; Periodic boundary conditionspbc         = xyz
; 3-D PBC; Dispersion correctionDispCorr    = EnerPres  ; account for cut-off
vdW scheme; Velocity generationgen_vel     = yes       ; assign velocities
from Maxwell distributiongen_temp    = 300       ; temperature for Maxwell
distributiongen_seed    = -1        ; generate a random seed
```

Saveing MD parameter files

Step 23.

Add the following lines into "**npt.mdp**" and save in the project folder.

```
title       = Protein-ligand complex NPT equilibration define      = -DPOSRES
; position restrain the protein and ligand; Run parametersintegrator  = md
; leap-frog integratornsteps      = 150000     ; 2 * 50000 = 100 psdt
= 0.002     ; 2 fs; Output controlnstxout     = 500       ; save coordinates
every 1.0 psnstvout     = 500       ; save velocities every 1.0 psnstenergy
= 500       ; save energies every 1.0 psnstlog      = 500       ; update log
file every 1.0 psenergygrps  = Protein ; Bond parameterscontinuation    = yes
; first dynamics runconstraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds)
constrainedlincs_iter      = 1             ; accuracy of LINCSlincs_order
= 4             ; also related to accuracy; Neighborsearchingcutoff-scheme
= Verletns_type         = grid      ; search neighboring grid cellsnstlist
= 10        ; 20 fs, largely irrelevant with Verletrcoulomb        = 1.4
; short-range electrostatic cutoff (in nm)rvdw            = 1.4       ;
short-range van der Waals cutoff (in nm); Electrostaticscoulombtype     = PME
; Particle Mesh Ewald for long-range electrostaticspme_order       = 4
; cubic interpolationfourierspacing  = 0.16      ; grid spacing for FFT;
Temperature couplingtcoupl      = V-rescale                    ; modified
Berendsen thermostattc-grps     = Protein_LIG Water_and_ions    ; two
coupling groups - more accuratetau_t       = 0.1   0.1                    ;
time constant, in psref_t       = 300   300                    ; reference
temperature, one for each group, in K; Pressure couplingpcoupl      =
Parrinello-Rahman             ; pressure coupling is on for NPTpcoupltype  =
isotropic                     ; uniform scaling of box vectorstau_p       =
2.0                           ; time constant, in psref_p       = 1.0
; reference pressure, in barcompressibility = 4.5e-5                    ;
isothermal compressibility of water, bar^-1refcoord_scaling    = com;
Periodic boundary conditionspbc         = xyz       ; 3-D PBC; Dispersion
correctionDispCorr    = EnerPres  ; account for cut-off vdW scheme; Velocity
generationgen_vel     = no        ; velocity generation off after NVT
```

Saveing MD parameter files

Step 24.

Add the following lines into "**md.mdp**" and save in the project folder.

```
title       = Protein-ligand complex MD simulation
; Run parameters
integrator  = md        ; leap-frog integrator
nsteps      = 1000000    ; 2 * 500000 = 1000 ps (1 ns)
dt          = 0.002     ; 2 fs
; Output control
nstxout             = 0         ; suppress .trr output
nstvout             = 0         ; suppress .trr output
nstenergy           = 5000      ; save energies every 10.0 ps
nstlog              = 5000      ; update log file every 10.0 ps
nstxout-compressed  = 5000      ; write .xtc trajectory every 10.0 ps
compressed-x-grps   = System
energygrps          = Protein
; Bond parameters
continuation    = yes           ; first dynamics run
constraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds)
constrained
lincs_iter      = 1             ; accuracy of LINCS
lincs_order     = 4             ; also related to accuracy
; Neighborsearching
cutoff-scheme   = Verlet
ns_type         = grid      ; search neighboring grid cells
nstlist         = 10        ; 20 fs, largely irrelevant with Verlet
rcoulomb        = 1.4       ; short-range electrostatic cutoff (in nm)
rvdw            = 1.4       ; short-range van der Waals cutoff (in nm)
; Electrostatics
coulombtype     = PME       ; Particle Mesh Ewald for long-range
electrostatics
pme_order       = 4         ; cubic interpolation
fourierspacing  = 0.16      ; grid spacing for FFT
; Temperature coupling
tcoupl      = V-rescale                     ; modified Berendsen thermostat
tc-grps     = Protein_LIG Water_and_ions    ; two coupling groups - more
accurate
tau_t       = 0.1   0.1                     ; time constant, in ps
ref_t       = 300   300                     ; reference temperature, one for
each group, in K
; Pressure coupling
pcoupl      = Parrinello-Rahman             ; pressure coupling is on for NPT
pcoupltype  = isotropic                     ; uniform scaling of box vectors
tau_p       = 2.0                           ; time constant, in ps
ref_p       = 1.0                           ; reference pressure, in bar
compressibility = 4.5e-5                    ; isothermal compressibility of
water, bar^-1
```

```
; Periodic boundary conditions
pbc         = xyz       ; 3-D PBC
; Dispersion correction
DispCorr    = EnerPres  ; account for cut-off vdW scheme
; Velocity generation
gen_vel     = no        ; assign velocities from Maxwell distribution
```

Saveing MD parameter files

Step 25.

Add the following lines into **"em_real.mdp"** and save in the project folder.

```
; LINES STARTING WITH ';' ARE COMMENTS
title           = Minimization        ; Title of run

; Parameters describing what to do, when to stop and what to save
integrator      = steep                ; Algorithm (steep = steepest descent
minimization)
emtol           = 1000.0          ; Stop minimization when the maximum force
< 10.0 kJ/mol
emstep      = 0.01      ; Energy step size
nsteps          = 50000                 ; Maximum number of (minimization)
steps to perform
energygrps      = Protein         ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to
calculate the interactions
nstlist             = 1                   ; Frequency to update the neighbor
list and long range forces
cutoff-scheme   = Verlet
ns_type             = grid               ; Method to determine neighbor list
(simple, grid)
rlist               = 1.0               ; Cut-off for making neighbor list
(short range forces)
coulombtype         = PME               ; Treatment of long range
electrostatic interactions
rcoulomb            = 1.0               ; long range electrostatic cut-off
rvdw                = 1.0               ; long range Van der Waals cut-off
pbc                     = xyz                ; Periodic Boundary Conditions
```

Adding ions

Step 26.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx grompp –f em.mdp –c solv.gro –p topol.top –o ions.tpr
```

Adding ions

Step 27.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx genion –s ions.tpr –o solv_ions.gro –p topol.top –neutral
```

Energy minimization

Step 28.

cmd COMMAND (Linux - 15.0.4)

```
$ gmx grompp –f em_real.mdp –c solv_ions.gro –p topol.top –o em.tpr
$ gmx mdrun –v –deffnm em
```

Restraining the ligand

Step 29.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx genrestr –f drug-atb.pdb –o posre_drug.itp –fc 1000 1000 1000
```

Restraining the ligand

Step 30.

Open '**topol.top**'.

And the following lines after '#include 'drug.itp'.

```
; Include Position restraint file#ifdef POSRES#include 'posre_drug.itp'#endif
```

```
; Include chain topologies
#include "topol_Protein_chain_A.itp"
#include "topol_Protein_chain_A2.itp"
#include "topol_Protein_chain_B.itp"
#include "topol_Protein_chain_B2.itp"

; Include drug topology
#include "drug.itp"

; Include Position restraint file
#ifdef POSRES
#include "posre_drug.itp"
#endif

; Include water topology
#include "gromos53a6_atb.ff/spc.itp"

#ifdef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
;  i funct       fcx        fcy        fcz
   1    1       1000       1000       1000
#endif

; Include topology for ions
```

Thermostats

Step 31.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx make_ndx –f em.gro –o index.ndx
```

SAFETY INFORMATION

Check your MD parameters file (MDP) for the line: tc_grps We make a group for protein_drug and solvent_ions. So check and you must edit nvt.mdp, npt.mdp, and md.mdp to have such as: tc-grps = Protein_LIG Water_and_ions ; two coupling groups - more accurate

Thermostats

Step 32.

Merge the protein and drug group:

Chose and type the following group in the make_ndx prompt:

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
> 1 | 13
> q
```

Contant Number of atoms, Volume, and Temprature (NVT) equilibrium.

Step 33.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx grompp –f nvt.mdp –c em.gro –p topol.top –n index.ndx –o nvt.tpr
$ gmx mdrun –v –deffnm nvt
```

SAFETY INFORMATION

In case you got error. Check your grouping in mdp files.

Contant Number of atoms, pressure, and Temprature (NPT) equilibrium (2).

Step 34.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx grompp –f npt.mdp –c nvt.gro –t nvt.cpt –p topol.top –n index.ndx –o nvt.tpr
$ gmx mdrun –v –deffnm npt
```

Production MD

Step 35.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx grompp –f md.mdp –c npt.gro –t npt.cpt –p topol.top –n index.ndx –o md.tpr
$ gmx mdrun –v –deffnm md
```

You can set you simulation time (e.g. 5ns or 10ns) in md.mdp file. (line: nsteps)

MD extension

Step 36.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx convert-tpr –s md.tpr –extend 3000 –e md.edr –o md_5.tpr
$ mdrun –v –s md_5.tpr –cpi md.cpt
```

For example if you already ran 3ns and stopped. Now you can extend MD into 5ns or more. Set your new output file (-o). Set your previous check point time. Be careful. The file names are important here. Example command is for extending into more 3ns from previous 2ns so we have here: -extend 3000 . The unit is ps.

RMSD computation

Step 37.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx rms –s md_10.tpr –f traj_comp.xtc –o rmsd_10.xvg
```

The file names are based on your previous MD production.

RMSF computation

Step 38.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ rmsf –s md_10.tpr –f traj_comp.xtc –o rmsf_10.xvg –oq rmsf_10.pdb
```

RDF computation

Step 39.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx rdf –s md_10.tpr –com –f traj_comp.xtc –o rdf_10.xvg
```

Radius of Gyration

Step 40.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx gyrate –s md_10.tpr –f traj_comp.xtc –o gyr_10.xvg
```

Hydrogen bonds

Step 41.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx hbond –s md_10.tpr –f traj_comp.xtc -num
```

Energy, enthalpy, enthropy, etc. computations

Step 42.

cmd COMMAND (Linux - Ubuntu 15.0.4)

```
$ gmx energy –s md_12.tpr –o energy.xvg
```

Charts

Step 43.

Open Grace to generate charts using generated xvg files.

Warnings

target_docked.pdb (e.g., *3og7_bbr_7.pdb*) already prepared by molecular docking.

Coordinates of ligand copied and cut into a new file, and saved them as *drug.pdb* and *target.pdb*. Therefore, two files were prepared for ligand and target, respectively.