Generation of topology files of a protein chain and simulations of a dipeptide

After studying this chapter, you will be able to

1) Generate a topology file of a dipeptide molecule.
2) Generate a topology file of a protein molecule.
3) Estimate the radius of gyration of the backbone of a dipeptide as a function of time.
4) Calculate the Root Mean Square Deviations (RMSDs) of the backbone of a dipeptide as a function of time.

Topology file of dipeptide

Generation of .pdb file of dipeptide using chemira

First of all, you have to open a chemira window. Click on tools → structure Editing → Build structure. After that, the following window will open (Fig. 36.1). You can see in this window that there many options like atom, fragment, peptide sequence etc.

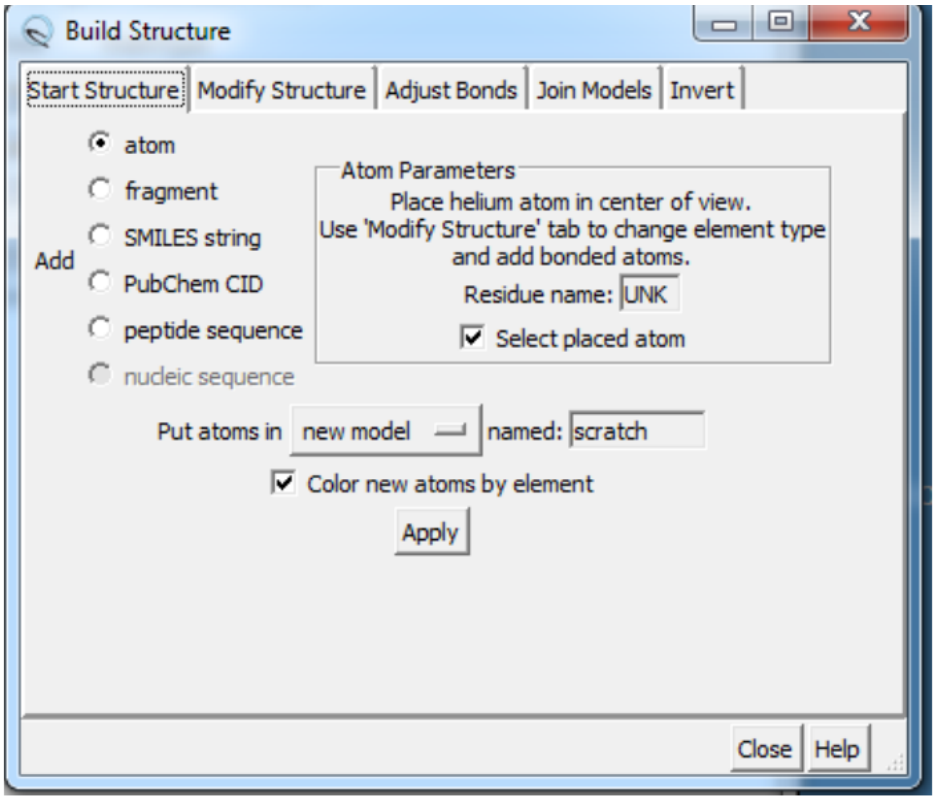

Fig 36.1 Display of chemira.

Now click on peptide sequence. Then you will see the following window (Fig. 36.2).

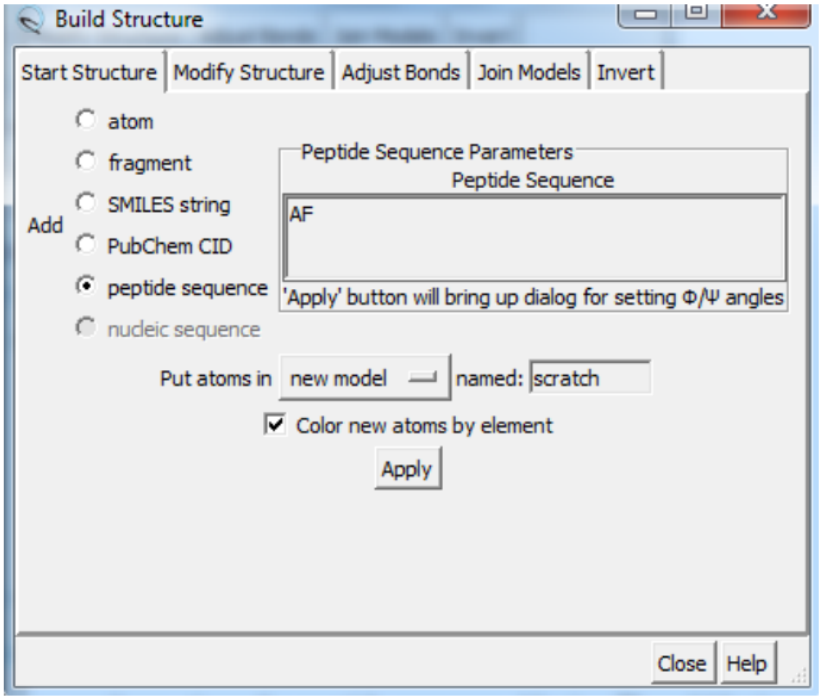

Fig. 36.2 Browser of chemira.

Here you can see that we have given the peptide sequence AF [Alanine (A)-PhenylAlanine (F)]. Click on the apply option. You will see the following window (Fig. 36.3)

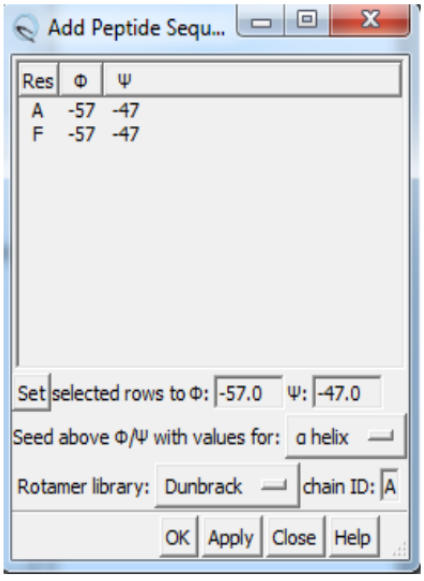

Fig.36.3 The selection of the sequence of amino acids has been shown.

In the above menu, you can adjust the angles (φ, ψ) of the dipeptide suitably. After adjusting the angles, click on the OK option at the bottom of the screen and then you will see the structure of the dipeptide (Fig. 36.4).

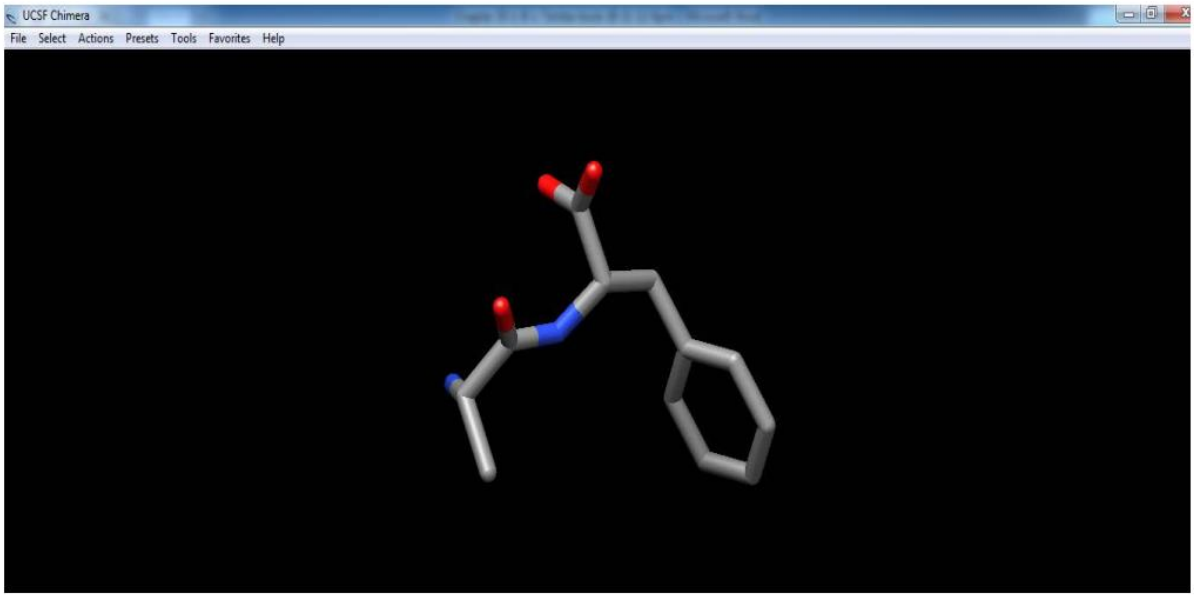

Fig. 36.4 Snapshot of dipeptide in VMD.

Now we will show you how to save this pepetide file in **.pdb** format. Click on file menu of the above Chemira window and then click on save PDB. Now you can save this in **.pdb** format on your desktop. The name of this file is **alaphe.pdb.**

Now make a directory dipeptide in ***'cd home/user/gromacs-4.5.4/gro4.5.4'*** by typing

makedir dipeptide

[press <enter> after every command]

Now go to the dipeptide directory by typing,

cd home/user/gromacs-4.5.4/gro4.5.4/dipeptide

Now copy the file **alaphe.pdb** (which you have saved in your Desktop) into **dipeptide** directory. Then go to the dipeptide directory and give the following command to generate the topology file.

pdb2gmx -f alaphe.pdb

On entering the above command, you will see:

Select the Force Field:

From '/usr/local/gromacs/share/gromacs/top':

1: AMBER03 force field (Duan et al., J. Comp. Chem. 24, 1999-2012, 2003)

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

3: AMBER96 force field (Kollman et al., Acc. Chem. Res. 29, 461-469, 1996)

4: AMBER99 force field (Wang et al., J. Comp. Chem. 21, 1049-1074, 2000)

5: AMBER99SB force field (Hornak et al., Proteins 65, 712-725, 2006)

6: AMBER99SB-ILDN force field (Lindorff-Larsen et al., Proteins 78, 1950-58, 2010)

7: AMBERGS force field (Garcia & Sanbonmatsu, PNAS 99, 2782-2787, 2002)

8: CHARMM27 all-atom force field (with CMAP) - version 2.0

9: GROMOS96 43a1 force field

10: GROMOS96 43a2 force field (improved alkane dihedrals)

11: GROMOS96 45a3 force field (Schuler JCC 2001 22 1205)

12: GROMOS96 53a5 force field (JCC 2004 vol 25 pag 1656)

13: GROMOS96 53a6 force field (JCC 2004 vol 25 pag 1656)

14: OPLS-AA/L all-atom force field (2001 aminoacid dihedrals)

15: [DEPRECATED] Encad all-atom force field, using full solvent charges

16: [DEPRECATED] Encad all-atom force field, using scaled-down vacuum charges

17: [DEPRECATED] Gromacs force field (see manual)

18: [DEPRECATED] Gromacs force field with hydrogens for NMR

For selecting **GROMOS96 53a6** force field type

13

and press <enter>

You will then be prompted to select the water model through the following sequence.

To Select the Water Model:

1: SPC simple point charge, recommended

2: **SPC/E** extended simple point charge

3: None

Type

2

press <enter>

to select the SPC/E model of water.

You have now successfully generated a topology from: **alaphe.pdb.**

The **Gromos53a6 force field** and the s**pce** water model are used.

The topology file of alaphe.pdb is **topol.top**. And initial structure file is **conf.gro.** The solvent structure file is **spce216.gro.** These three files are generated by the above sequence of operations.

Defining the simulation box and solvating the dipeptide with water

Input files: conf.gro, topol.top, spce216.gro

You can define the cubic box around the dipeptide by using editconf commond

editconf -f conf.gro -o simulationbox.gro -c -d 1.0 -bt cubic

press <enter>

We have defined a simulation box by the command given above. We can solvate the dipeptide with solvent (water) by using the following genbox command:

genbox -cp simulationbox.gro -cs spce216.gro -o dipep_solv.gro -p topol.top

press <enter>

Adding ions to neutralize the dipeptide

We now have a solvated dipeptide. The output of pdb2gmx told us that the dipeptide has a net charge of +2*e* (based on its amino acid composition). If you missed this information in the pdb2gmx output, look at the last line of your [atoms] directive in topol.top; it should read (in part) "qtot 8." Since the system should be electrically neutral without a net charge, we must add counter ions to our system to maintain electroneutrality.

To neutralize the charged dipeptide, we can use the tool **genion** for adding ions within GROMACS. What genion does is to read through the topology and replace water molecules with the ions that the user specifies.

To produce a .tpr file with grompp, we will need an additional input file, with the extension .mdp (molecular dynamics parameter file); grompp will assemble the parameters specified in the .mdp file with the coordinates and topology information to generate a .tpr file. An .mdp file is normally used to run energy minimization or an MD simulation, but in this case, it is simply used to generate an atomic description of the system. An example, ions.mdp file (the one we will use) is given below:

****

```
; ions.mdp - used as input into grompp to generate ions.tpr
; 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 <
1000.0 kJ/mol/nm
emstep          = 0.01          ; Energy step size
nsteps         = 50000        ; Maximum number of (minimization) steps to
perform

; 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



 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          ; Short-range electrostatic cut-off
 rvdw           = 1.0          ; Short-range Van der Waals cut-off
 pbc            = xyz          ; Periodic Boundary Conditions (yes/no)
```

****

To neutralize the system, use the following command.

grompp -f ions.mdp -c dipep_solv.gro -p topol.top -o ions.tpr

Now we have an atomic-level description of our system in the binary file ions.tpr. We will pass this file to genion by typing the following command.

genion -s ions.tpr -o dipep_solv_ions.gro -p topol.top -pname NA -nname CL -nn 2

press <enter>

You can choose group "SOL" for embedding ions and replacing water molecules. You do not want to replace parts of your dipeptide with ions.

The directive in topol.top file [molecules] should now look like:

```
[ molecules ]
; Compound       #mols
Protein_A          1
SOL              216
CL                 2
```

Energy Minimization (EM) of Dipeptide solvated with water molecules

Now we have solvated and electroneutralized the dipeptide water system. Before we can begin dynamics, we must ensure that the system has no steric repulsions or inappropriate geometries. We will minimize the energy of the system by using the steepest (descent) algorithm. The em.mdp (the file containing the parameters for energy minimization) file is given below.

```
;******************** em.mdp file ***************************************

; minim.mdp - used as input into grompp to generate em.tpr
; 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 <
1000.0 kJ/mol/nm
emstep          = 0.01          ; Energy step size



nsteps          = 50000         ; Maximum number of (minimization) steps to
perform

; 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
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           ; Short-range electrostatic cut-off
rvdw            = 1.0           ; Short-range Van der Waals cut-off
pbc             = xyz           ; Periodic Boundary Conditions (yes/no)


;***************************end of em.mdp file*********************************
```

The energy minimization is accomplished by the following two commands.

grompp -f em.mdp -c dipep_solv_ions.gro -p topol.top -o em.tpr

press <enter>

mdrun -v -deffnm em

press <enter>

Note: The ***-v*** flag is for the impatient: it makes mdrun verbose, such that it prints its progress to the screen at every step. The ***-deffnm*** flag will define the file names of the input and output.

There are two very important factors to evaluate to determine if EM is successful. The first is the potential energy (printed at the end of the EM process, even without -v). E_{pot} should be negative, and (for a simple protein in water) of the order of 10^5-10^6 $kJmol^{-1}$, depending on the system size and the number of water molecules. The second important feature is the maximum force, F_{max}, the target for which was set in minim.mdp - "emtol = 1000.0" - indicating a target F_{max} of no greater than 1000 kJ mol^{-1} nm^{-1}. It is possible to arrive at values of E_{pot} with F_{max} > emtol. If this happens, your system may not be stable enough for simulation. Evaluate why it may be happening, and perhaps change your minimization parameters (integrator, emstep, etc).

Let's do a bit of analysis. The em.edr file contains all of the energy terms that GROMACS collects during EM. You can analyze any .edr file using the GROMACS tools g_energy:

g_energy -f em.edr -o potential.xvg

At the prompt, type "10 0" to select Potential (10); zero (0) terminates input. You will be shown the average of E_{pot}, and a file called "potential.xvg" will be written. To plot this data, you will need the Xmgrace plotting tool.

Equilibration of Energy Minimized solvated and neutralized dipeptide.

EM ensured that we have a reasonable starting structure, in terms of geometry and solvent orientation. To begin real dynamics, we must equilibrate the solvent and ions around the protein.

Equilibration is often conducted in two phases. The first phase is conducted under an *NVT* ensemble (constant Number of particles, Volume, and Temperature). Typically, 50-100 ps should suffice, and we will conduct a 100-ps *NVT* equilibration for this exercise. The .mdp file is given below

**************************nvt.mdp file**

```
title           = Dipeptide NVT equilibration
define          = -DPOSRES      ; position restrain the protein
; Run parameters
integrator      = md            ; leap-frog integrator
nsteps          = 50000         ; 2 * 50000 = 100 ps
dt              = 0.002         ; 2 fs
; Output control
nstxout         = 100           ; save coordinates every 0.2 ps
nstvout         = 100           ; save velocities every 0.2 ps
nstenergy       = 100           ; save energies every 0.2 ps
nstlog          = 100           ; update log file every 0.2 ps
; Bond parameters
continuation    = no            ; 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
ns_type         = grid          ; search neighboring grid cells
nstlist         = 5             ; 10 fs
rlist           = 1.0           ; short-range neighborlist cutoff (in nm)
rcoulomb        = 1.0           ; short-range electrostatic cutoff (in nm)
rvdw            = 1.0           ; 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 is on
tcoupl          = V-rescale     ; modified Berendsen thermostat
tc-grps         = Protein Non-Protein  ; 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 is off
pcoupl          = no            ; no pressure coupling in NVT
; Periodic boundary conditions
pbc             = xyz           ; 3-D PBC
; Dispersion correction
DispCorr        = EnerPres      ; account for cut-off vdW scheme
; Velocity generation
gen_vel         = yes           ; assign velocities from Maxwell distribution
gen_temp        = 300           ; temperature for Maxwell distribution
```

```
gen_seed       = -1           ; generate a random seed
```
**********end of nvt.mdp file**********

Now we will run grompp and mdrun command.

grompp -f nvt.mdp -c em.gro -p topol.top -o nvt.tpr

<press enter>

mdrun -deffnm nvt

press <enter>

We can analyze the temperature progression, by using g_energy:

g_energy -f nvt.edr

Type "15 0" to select the temperature of the system and exit.

Performance of NPT equilibration of NVT equilibrated dipeptide.

Now we are going to perform NPT equilibration of NVT equilibrated dipeptide. The npt.mdp file is given below.

**********npt.mdp**********

```
title          = equilibrated dipeptide NPT equilibration
define         = -DPOSRES     ; position restrain the protein
; Run parameters
integrator     = md           ; leap-frog integrator
nsteps         = 50000        ; 2 * 50000 = 100 ps
dt             = 0.002        ; 2 fs
; Output control
nstxout        = 100          ; save coordinates every 0.2 ps
nstvout        = 100          ; save velocities every 0.2 ps
nstenergy      = 100          ; save energies every 0.2 ps
nstlog         = 100          ; update log file every 0.2 ps
; Bond parameters
continuation   = yes          ; Restarting after NVT
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
ns_type        = grid         ; search neighboring grid cells
nstlist        = 5            ; 10 fs
rlist          = 1.0          ; short-range neighborlist cutoff (in nm)
rcoulomb       = 1.0          ; short-range electrostatic cutoff (in nm)
rvdw           = 1.0          ; 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 is on
tcoupl          = V-rescale     ; modified Berendsen thermostat
tc-grps         = Protein Non-Protein  ; 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 is on
pcoupl          = Parrinello-Rahman    ; Pressure coupling on in 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
refcoord_scaling = com
; Periodic boundary conditions
pbc             = xyz           ; 3-D PBC
; Dispersion correction
DispCorr        = EnerPres      ; account for cut-off vdW scheme
; Velocity generation
gen_vel         = no            ; Velocity generation is off
```

**********end of full.mdp file**********

We will run grompp and mdrun commands.

grompp -f full.mdp -c nvt.gro -p topol.top -o npt.tpr

mdrun -deffnm npt

Now we will analyze the progress of the pressure of the system by using command g_energy.

g_energy -f npt.edr -o pressure.xvg

Type "16 0" to select the pressure of the system and exit.

Performance of full molecular dynamics simulation of equilibrated dipeptide.

We have now a properly equilibrated the system at room temperature and pressure. We will run 1ns molecular dynamics simulation of the system. The required full.mdp file is given below

**********full.mdp file**********

```
title           = OPLS Lysozyme MD
; Run parameters
integrator      = md            ; leap-frog integrator
nsteps          = 500000        ; 2 * 500000 = 1000 ps, 1 ns
dt              = 0.002         ; 2 fs
; Output control
nstxout         = 1000          ; save coordinates every 2 ps
nstvout         = 1000          ; save velocities every 2 ps
```

```
nstxtcout       = 1000          ; xtc compressed trajectory output every 2 ps
nstenergy       = 1000          ; save energies every 2 ps
nstlog          = 1000          ; update log file every 2 ps
; Bond parameters
continuation    = yes           ; Restarting after NPT
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
ns_type         = grid          ; search neighboring grid cells
nstlist         = 5             ; 10 fs
rlist           = 1.0           ; short-range neighborlist cutoff (in nm)
rcoulomb        = 1.0           ; short-range electrostatic cutoff (in nm)
rvdw            = 1.0           ; 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 is on
tcoupl          = V-rescale     ; modified Berendsen thermostat
tc-grps         = Protein Non-Protein  ; 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 is on
pcoupl          = Parrinello-Rahman    ; Pressure coupling on in 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            ; Velocity generation is off
```

**********end of full.mdp file**********

We are going to run grompp and mdrun command.

grompp -f md.mdp -c npt.gro -p topol.top -o dipep_full_1ns

mdrun -deffnm dipep_full_1ns

Analysis of the simulated dipeptide.

We have simulated the dipeptide. We will show some analysis of the dipeptide.

Calculation of the radius of gyration and MSDs of dipeptide.

The radius of gyration (R_g) of a protein is a measure of its compactness. If a protein is stably folded, it is likely to maintain a relatively steady value of R_g. If a protein unfolds, its R_g will change over time. Let us analyze the radius of gyration for the dipeptide in our simulation using:

g_gyrate -s dipep_full_1ns.tpr -f dipep_full_1ns.xtc -o gyrate.xvg

<enter>

Type "4" to select the backbone of dipeptide.

<enter>

The file gyrate,xvg will be generated. Use xmgrace to plot the graph between radius of gyration and time for the backbone of the dipeptide.

Xmgrce gyrate.xvg

<enter>

You will see the following type of plot (Fig. 36.5).

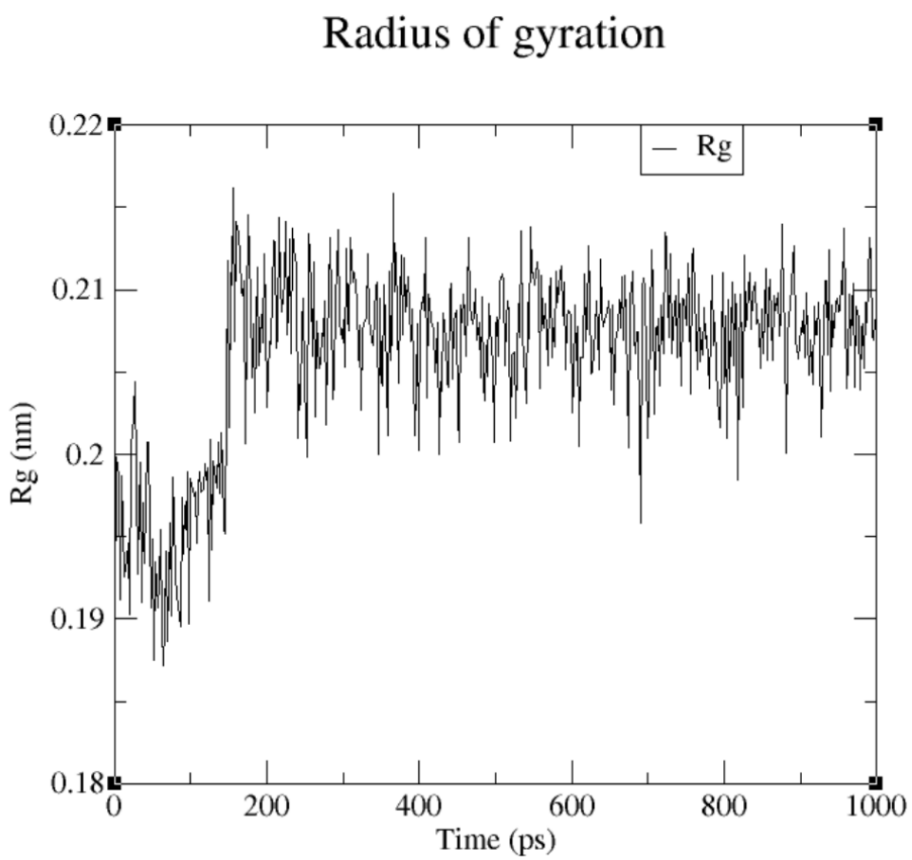

Fig. 36.5 Radius of gyration vs. time plot for the backbone of the dipeptide.

The small value of R_g of 2.0 Å indicates a compact structure of the dipeptide.

The root mean square deviations (MSDs) of a dipeptide or a protein display the diffusive behavior of these molecules. Let us display the computed MSDs for the backbone of the dipeptide in our simulation using:

g_msd -s dipep_full_1ns.tpr -f dipep_full_1ns.xtc -o msd.xvg

<enter>

Type "4" to select the backbone of dipeptide.

<enter>

The file msd,xvg will be generated. Use xmgrace to plot the graph between MSDs and time for the backbone of dipeptide.

Xmgrce msd.xvg

<enter>

You can see the following type of plot (Fig. 36.6).

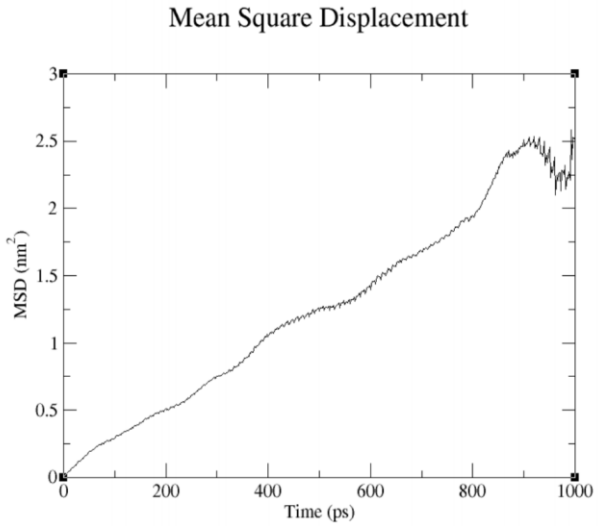

Fig. 36.6 Mean square displacements vs. time plot for the backbone of dipeptide.

From the slope of the nearly linear MSDs with time, the value of the diffusion constant of the dipeptide is obtained. In the above case, D is 0.42×10^{-5} $cm^2 sec^{-1}$.

Summary

In this chapter we have generated the topology file of a dipeptide and then the dipeptide has been solvated with water molecules. The energy minimization of solvated dipeptide has been done. We have equilibrated the dipeptide with NVT and NPT ensembles. Finally we have performed the molecular dynamics simulation of an equilibrated dipeptide. We have also calculated the radius of gyration and the diffusion constant of the dipeptide AF [Alanine (A)-PhenylAlanine (F)].

Assignments

[1] Analyze the RMSD of backbone of dipeptide.
[2] Make the topology files of another dipeptide, a tripeptide and a polypeptide (with four amino acids residues) and perform molecular dynamics simulations.
[3] Calculate the density of the system by using the command

g_energy -f npt.edr -o density.xvg