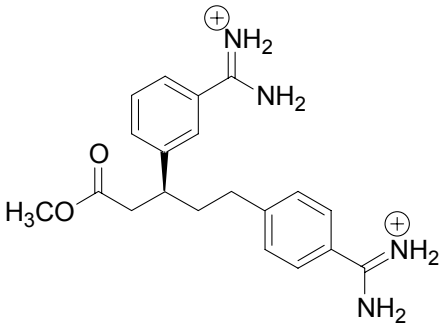
GROMACS Tutorial for Drug – Enzyme Complex.

John E. Kerrigan, Ph.D.
IST/Academic Computing Services
RWJMS – University of Medicine and Dentistry of NJ
675 Hoes Lane, RM SB11
Piscataway, NJ 08854

Phone: (732) 235-4473
Fax: (732) 235-5252
Email: kerrigje@umdnj.edu
Webpage: http://www2.umdnj.edu/~kerrigje

```
HEADER    SERINE PROTEASE                         26-NOV-97   1AZ8
TITLE     BOVINE TRYPSIN COMPLEXED TO BIS-PHENYLAMIDINE INHIBITOR
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: TRYPSIN;
COMPND   3 CHAIN: NULL;
COMPND   4 EC: 3.4.21.4
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: BOS TAURUS;
SOURCE   3 ORGANISM_COMMON: BOVINE;
SOURCE   4 ORGAN: PANCREAS
KEYWDS    HYDROLASE, SERINE PROTEINASE
EXPDTA    X-RAY DIFFRACTION
AUTHOR    R.ALEXANDER,A.SMALLWOOD
REVDAT   1   13-JAN-99 1AZ8    0
JRNL        AUTH   R.ALEXANDER,T.MADUSKUIE,K.MCNAMARA,A.SMALLWOOD,
JRNL        AUTH 2 A.WEI,P.STOUTEN
JRNL        TITL   UNEXPECTED BINDING MODE OF A BIS-PHENYLAMIDINE
JRNL        TITL 2 FACTOR XA INHIBITOR COMPLEXED TO BOVINE TRYPSIN
JRNL        REF    TO BE PUBLISHED
```

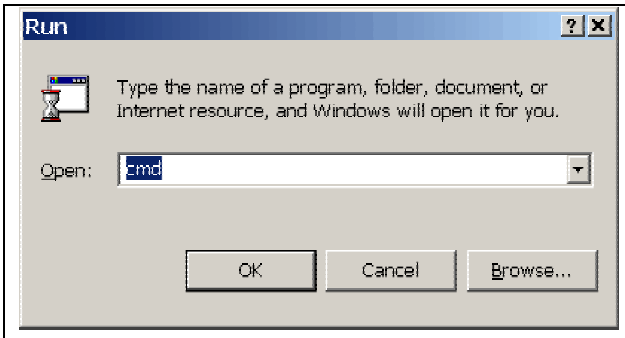


Note: You will generate gromacs (*.gro) structure files in this tutorial. To view these files, you must use either gOpenMol (Download from: http://staff.csc.fi/~laaksone/gopenmol/gopenmol.html) or VMD (Download from: http://www.ks.uiuc.edu/Research/vmd/).

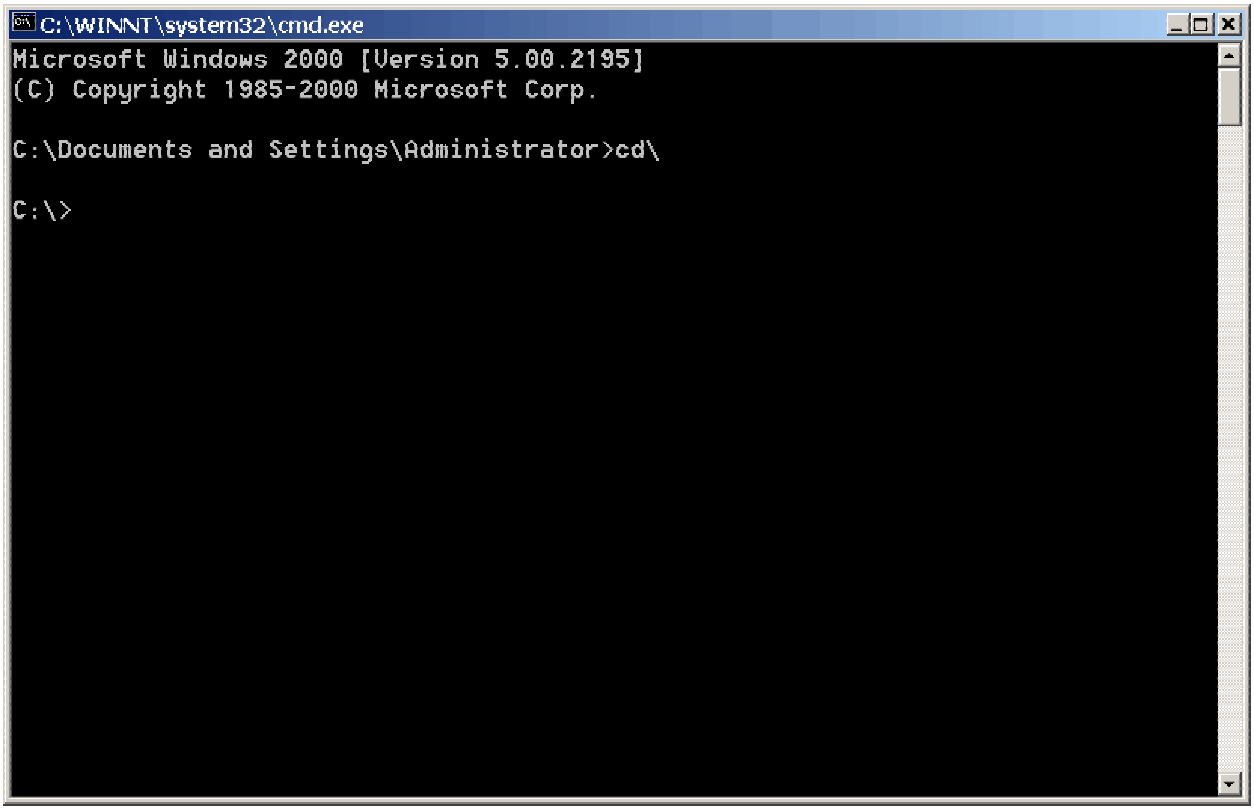
Gromacs is a very powerful molecular simulation package.[1] Like other molecular mechanics/dynamics software packages, Gromacs uses an internal set of databases, which contain parameters for the amino acids, nucleic acids, cofactors, and some lipids that may be present in your PDB file. Gromacs does not know how to parameterize anything else in your PDB file (i.e. the HETATM records). You must provide these missing parameters. In this tutorial, you will learn how to use the PRODRG2 server to parameterize your HETATM group (the drug) and include it into your calculations.

NOTE: This tutorial assumes that you are using Gromacs in a linux/unix environment. If you are using the Gromacs port to Windows, you will need to run the commands that follow from within the Windows command shell (formerly dosshell).

Go To **Start > Run** and type "cmd" in the Run dialog to run the command shell.



Click OK



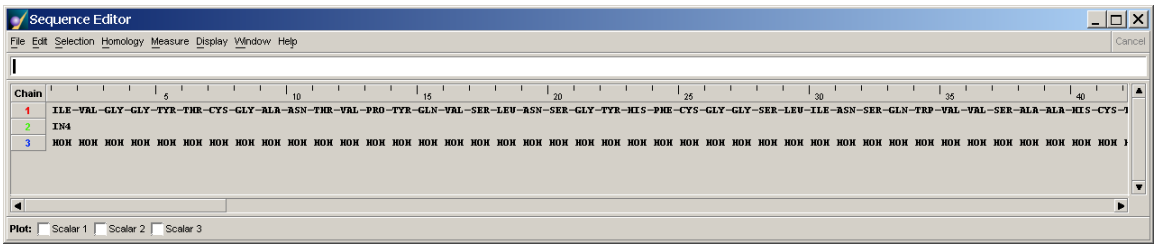
The Windows 2000 command shell.

Establish a working directory for your project. For example, to make a project directory in your zip drive (assuming drive letter D: on some systems the zip drive is assigned drive letter "E"). This may also be accomplished using Windows Explorer (use the 'MS window key – E key' combination to launch Explorer).

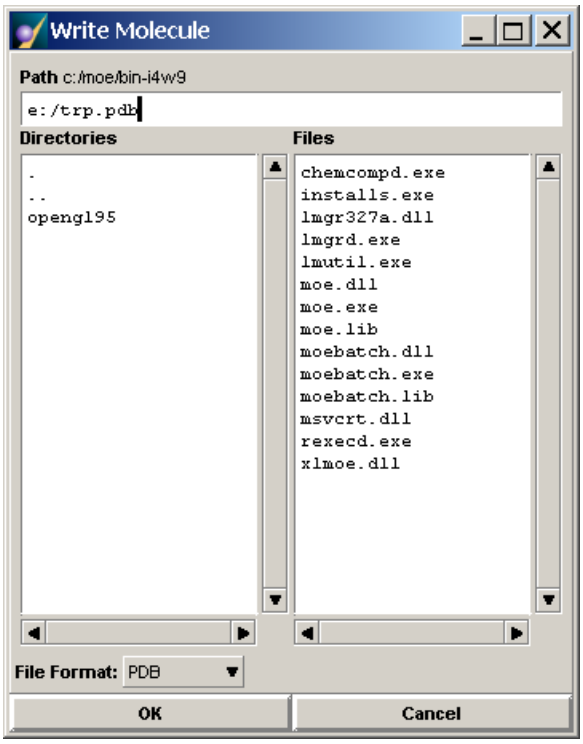
cd D:
mkdir project
cd project

In linux/unix, establish a project directory. Use mkdir project (same command as in Windows). Then cd in to that directory and perform all of your work in your project directory.

Open 1AZ8.pdb in MOE. Open Sequence Editor and delete the water chain (chain #3) and the drug chain (IN4; chain #2). With your mouse click on chain #2 to select it. Go to Edit > Delete Selected Chains. Perform the same operation to delete chain #3.



Save the remaining protein as trp.pdb. Use File > Save As…



File > Close in MOE

File > Open > 1AZ8.pdb

Now open the sequence editor and delete chains #1 and #3.

Save the drug as drg.pdb.
****

Next we will use the Dundee PRODRG[2] server to build a Gromacs 87 topology for the drug (drg.pdb).

In your browser, go to http://davapc1.bioch.dundee.ac.uk/programs/prodrg/prodrg.html and copy and paste (ctrl-v) your drg.pdb coordinates into the empty text box on the webpage. The view above is what you will see initially before pasting your coordinates. Check the following options:

Chirality	Yes
Full charges	Yes
Energy Minimization	No

Click “Run PRODRG”.

Download the Zipped Archive.

The 00README file has the following information about the files contained in the archive:

```
This is an explanation of what the output files mean/contain.
Not all files are listed, only the essential stuff.


DRGCNS.PAR  : CNS parameter file
DRGCNS.TOP  : CNS topology file
DRGDRG.MDS  : the PRODRG MOLDES (see paper)
DRGDRG.TOP  : the GROMOS87 topology building block
DRGFIN.GRO  : final coordinates (in GROMOS87 format, not PDB!) after EM
DRGFIN.PDB  : the final PDB file with all hydrogens added
DRGGMX.TOP  : GROMACS .itp file (NOTE: This is now correctly named
drggmx.itp)
DRGHEX.TOP  : HEX topology
DRGML2.TOP  : MOL2 file with only polar hydrogens
DRGML2.TOPH : MOL2 file with all hydrogens
DRGNOH.PDB  : the final PDB file without any hydrogens added
DRGPOH.PDB  : the final PDB file with only polar hydrogens added
DRGSHX.TOP  : SHELXL topology
DRGTRS.O    : O torsion database (read in with: read DRGTRS.O in O)
DRGWIF.TOP  : the WHAT IF topology
```

We want the DRGGMX.ITP file for building the topology for the drug. In addition, we will need the DRGFIN.GRO file for building the coordinate file (*.GRO). Extract the DRGPOH.PDB file for reference use.

Rename the DRGGMX.ITP file to drg.itp
Process the trp.pdb file with pdb2gmx.

pdb2gmx –f trp.pdb **–o** trp.gro **–p** trp.top

Select the Gromacs forcefield (Type “0” then hit <Enter>)

Next we must edit the drg.itp, trp.gro, and trp.top files!

Edit the drg.itp file. You should always edit the drg.itp file to make sure that the atom types and the charges make sense. (NOTE: See the Gromacs manual for a description of gromacs atom types.)

Here is the finished atoms section of our file:

```
[ atoms ]
 ;  nr  type resnr resid  atom  cgnr charge
     1    NT     1 IN4      N3     1    0.145
     2     H     1 IN4     HAA     1    0.072
     3     H     1 IN4     HAB     1    0.072
     4    CB     1 IN4     C20     1    0.422
```

```
 5    NT     1 IN4      N4     1    0.145
 6     H     1 IN4     HAF     1    0.072
 7     H     1 IN4     HAE     1    0.072
 8    CB     1 IN4      C6     2    0.000
 9  CR61     1 IN4      C5     2    0.000
10  CR61     1 IN4      C1     2    0.000
11  CR61     1 IN4      C2     2    0.000
12  CR61     1 IN4      C3     3    0.000
13    CB     1 IN4      C4     3    0.000
14   CH1     1 IN4     C10     3    0.000
15   CH2     1 IN4     C14     3    0.000
16     C     1 IN4     C22     4    0.469
17     O     1 IN4      O3     4   -0.576
18    OS     1 IN4      O4     4   -0.113
19   CH3     1 IN4      C8     4    0.089
20   CH2     1 IN4      C9     4    0.131
21   CH2     1 IN4     C11     5    0.000
22    CB     1 IN4     C12     5    0.000
23  CR61     1 IN4     C19     6    0.000
24  CR61     1 IN4     C17     6    0.000
25  CR61     1 IN4      C7     6    0.000
26  CR61     1 IN4     C13     6    0.000
27    CB     1 IN4     C15     6    0.000
28    CB     1 IN4     C16     7    0.423
29    NT     1 IN4      N1     7    0.144
30     H     1 IN4     HAH     7    0.072
31     H     1 IN4     HAG     7    0.072
32    NT     1 IN4      N2     7    0.145
33     H     1 IN4     HAD     7    0.072
34     H     1 IN4     HAC     7    0.072
```

No errors in atom typing were found. No charges were missing; however, always be sure to check for missing charges on heteroatoms. Notice that charges are localized into "groups". The "cgnr" field is the "charge group number". For comparison study, you should estimate charges using a quantum mechanics software package like Gaussian (www.gaussian.com) or Spartan (www.wavefun.com).

The itp file contains molecule information only. We are building this itp file from a "top" file, and will need to remove some "top" file lines (already present in the top file generated by pdb2gmx).

Open DRGFIN.GRO in a text editor (or view the contents in a separate unix shell using the "more" command).

Open trp.gro in a text editor (Word, vi, jot, etc.). Copy and paste the coordinates from DRFIN.GRO onto the end of the trp.gro file. Change the number at the top or beginning of the file from 2099 to 2133 to correct the total number of atoms in the file.

The drug will be residue #224 (after 223ASN in the trp.gro file). Change the residue name of the drug to 224IN4. Adjust the columns accordingly and renumber the atoms in a continuous fashion from the end of the original file (i.e. the first atom in the DRG should be atom # 2100, the second 2101, and so on…). Save trp.gro (overwrite the old file).

Edit trp.top using any text editor. Add **#include "drg.itp"** after the forcefield parameter include statement. Under the [molecules] section, add IN4 under Compound and 1 under #mols.

Setup the box

editconf –bt octahedron **–f** trp.gro **–o** trp.gro **–c –d** 0.5

What you have done in this command is specify that you want to use a truncated octahedral periodic box. The **–c** flag centers the molecule(s) in the box and the **–d** 0.5 flag sets the dimensions of the box based upon setting the box edge approx 0.5 nm from the molecule(s). Ideally you should set **–d** at no less than 0.5 nm to an upper limit of 1.0 nm or more depending upon the system. CAUTION: We are using as small a value as possible for this simulation. Normally, you would use a larger spacing.

genbox –cp trp.gro **–cs** spc216.gro **–o** trp_b4ion.gro **–p** trp.top

The genbox command generates the water box based upon the dimensions/box type that you had specified using editconf. In the command above, you are using the "SPC" water model. [3] The genbox program will add the correct number of water molecules needed to solvate your box of given dimensions. We are naming our output file using "b4ion". We will need to use genion later neutralize the charge in this system.

Before we can use genion, we must generate a tpr file for input. Therefore, we will setup the energy minimization, add ions and redo the energy min setup. Use the em.mdp file. Look into the contents of this file. It specifies a steepest descents minimization to remove bad van der Waals contacts. [LOCATION OF CPP ON WINDOWS MACHINES: On most systems this location is C:\gromacs\bin\cpp]

Content of em.mdp:

```
title               =  drg_trp
cpp                 =  /lib/cpp ; location of cpp on SGI
define              =  -DFLEX_SPC ; Use Ferguson's Flexible water model [4]
constraints         =  none
integrator          =  steep
dt                  =  0.002    ; ps !
nsteps              =  500
nstlist             =  10
ns_type             =  grid
rlist               =  0.9
coulombtype         =  PME ; Use particle-mesh ewald
rcoulomb            =  0.9
rvdw                =  1.0
fourierspacing      =  0.12
fourier_nx        =  0
fourier_ny        =  0
fourier_nz        =  0
pme_order         =  4
ewald_rtol        =  1e-5
```

```
optimize_fft      =  yes
;
;       Energy minimizing stuff
;
emtol               =  1000.0
emstep              =  0.01
```

grompp –f em.mdp **–c** trp_b4ion.gro **–p** trp.top **–o** trp_b4ion.tpr

You will notice that there is one warning noting that the system has a nonzero total charge (+9.0). We must neutralize this charge with 9 Cl- ions (for PME total charge must = ~0). We will use a program called genion to add the Cl ions.

genion –s trp_b4ion.tpr **–o** trp_b4em.gro **–nname** Cl **–nn** 9 **–g** trp_ion.log

You will be prompted to select a group of continuous water molecules. Select Group 13 (SOL).

You must make the following additions/corrections to your trp.top file:

Add the line

#include "ions.itp"

after the include statement for the forcefield or the drug. Subtract 9 from the total number of SOL molecules and add a line for 9 Cl molecules at the end of the trp.top file.

We must include the drug (IN4) and the ions (Cl) in the temperature coupling schemes in the pr.mdp and md.mdp files later on.

Run grompp to setup the energy minimization.

grompp –f em.mdp **–c** trp_b4em.gro **–p** trp.top **–o** trp_em.tpr

Run the energy minimization in background (Use the ampersand & at the end of the command). [WINDOWS USERS: You cannot run jobs in background in Windows. Omit the ampersand and nohup from the command below.]

nohup mdrun –s trp_em.tpr **–o** trp_em.trr **–c** trp_b4pr.gro **–g** em.log **–e** em.edr **&**

nohup is used to prevent hang-ups of background jobs on linux/unix systems.

Use the tail command to check on the progress of the minimization. Our run converged to 1000 in 335 steps. [WINDOWS USERS: You do not have the tail command. Use the type command instead. For example, use "type md.log" to view the contents of em.log.]

tail –15 em.log

Setup of the Position-Restrained Dynamics:

We will run a position-restrained dynamics simulation to "soak" the water and the drug into the drug-enzyme complex. In this run, we will restrain the atom positions of the protein to restrict their movement in the simulation (i.e. the atom positions are restrained not fixed!). The water and the drug will be permitted to relax about the protein. The relaxation time of water is 10 ps. Therefore, we will use a 20 ps dynamics run to perform the soak. Under coulombtype, PME stands for "Particle Mesh Ewald" electrostatics. PME is the best method for computing long-range electrostatics (gives more reliable energy estimates). [5, 6] The all-bonds option under constraints applies the Linear Constraint algorithm[7] for fixing all bond lengths (important to use this option when dt > 0.001 ps). We are using Berendsen's temperature and pressure coupling methods. [8]

```
title               =  trp_drg
warnings            =  10
cpp                 =  /lib/cpp ; location of cpp on SGI
define              =  -DPOSRES
constraints         =  all-bonds
integrator          =  md
dt                  =  0.002  ; ps !
nsteps              =  10000  ; total 20.0 ps.
nstcomm             =  1
nstxout             =  250 ; output coordinates every 0.5 ps
nstvout             =  1000 ; output velocities every 2.0 ps
nstfout             =  0
nstlog              =  10
nstenergy           =  10
nstlist             =  10
ns_type             =  grid
rlist               =  0.9
coulombtype         =  PME
rcoulomb            =  0.9
rvdw                =  1.0
fourierspacing      =  0.12
fourier_nx        =  0
fourier_ny        =  0
fourier_nz        =  0
pme_order         =  6
ewald_rtol        =  1e-5
optimize_fft      =  yes
; Berendsen temperature coupling is on in four groups
Tcoupl              =  berendsen
tau_t               =  0.1       0.1   0.1   0.1
tc_grps             =  protein   sol   IN4   Cl
ref_t               =  300       300   300   300
; Pressure coupling is on
Pcoupl              =  berendsen
pcoupltype          =  isotropic
tau_p               =  0.5
compressibility     =  4.5e-5
ref_p               =  1.0
; Generate velocites is on at 300 K.
gen_vel             =  yes
gen_temp            =  300.0
gen_seed            =  173529
```

grompp –f pr.mdp **–c** trp_b4pr.gro **–p** trp.top **–o** trp_pr.tpr

[WINDOWS USERS: You cannot run jobs in background in Windows. Omit the ampersand and nohup from the command below.]

nohup mdrun –s trp_pr.tpr **–o** trp_pr.trr **–c** trp_b4md.gro **–g** pr.log **–e** pr.edr **&**

Setting up the Molecular Dynamics Simulation:
The parameters shown below are for the Gromacs 87 force field, which we are using in this example. Consult the Gromacs manual for optimal settings for other force fields (e.g. the GROMOS 96 force field uses nstlist = 5 and rvdw = 1.4 is required).

Note: nstxout sets the snapshot frequency (e.g. nstxout = 500 writes out coordinates every 500 steps or 1.0 ps).

Content of md.mdp for NPT ensemble 50 ps simulation.

```
title               =  trp_drg MD
cpp                 =  /lib/cpp ; location of cpp on SGI
constraints         =  all-bonds
integrator          =  md
dt                  =  0.002  ; ps !
nsteps              =  25000  ; total 50 ps.
nstcomm             =  1
nstxout             =  500 ; output coordinates every 1.0 ps
nstvout             =  0
nstfout             =  0
nstlist             =  10
ns_type             =  grid
rlist               =  0.9
coulombtype         =  PME
rcoulomb            =  0.9
rvdw                =  1.0
fourierspacing      =  0.12
fourier_nx        =  0
fourier_ny        =  0
fourier_nz        =  0
pme_order         =  6
ewald_rtol        =  1e-5
optimize_fft      =  yes
; Berendsen temperature coupling is on in four groups
Tcoupl                =  berendsen
tau_t                 =  0.1        0.1   0.1   0.1
tc-grps               =  protein    IN4   sol   Cl
ref_t                 =  300        300   300   300
; Pressure coupling is on
Pcoupl              =  berendsen
pcoupltype          =  isotropic
tau_p               =  0.5
compressibility     =  4.5e-5
ref_p               =  1.0
; Generate velocites is on at 300 K.
gen_vel             =  yes
```

```
gen_temp            =  300.0
gen_seed            =  173529
```

grompp –f md.mdp **–c** trp_b4md.gro **–p** trp.top **–o** trp_md.tpr

[WINDOWS USERS: You cannot run jobs in background in Windows. Omit the ampersand and nohup from the command below.]

nohup mdrun –s trp_md.tpr **–o** trp_md.trr **–c** trp_pmd.gro **–g** md.log **–e** md.edr **&**

This run will take a long time (2 to 3 hours depending on the speed of your processor).

Use the **tail** command to check the md.log file. Or, if you are using the Windows command shell, use the "type" command.

When the simulation is finished, compress the trajectory using **trjconv**.

trjconv –f filename.trr **–o** filename.xtc

Remove the *.trr file to save on disk space (use **rm** trp_md.trr).

Use **ngmx** to view the trajectory (you may also download to your PC and use VMD to view the trajectory. ***Note***: VMD will ask for the trajectory file and the *.gro file for input.). (NOTE: ngmx does not run in Windows.)

ngmx –f trp_md.trr (or trp_md.xtc) **–s** trp_md.tpr

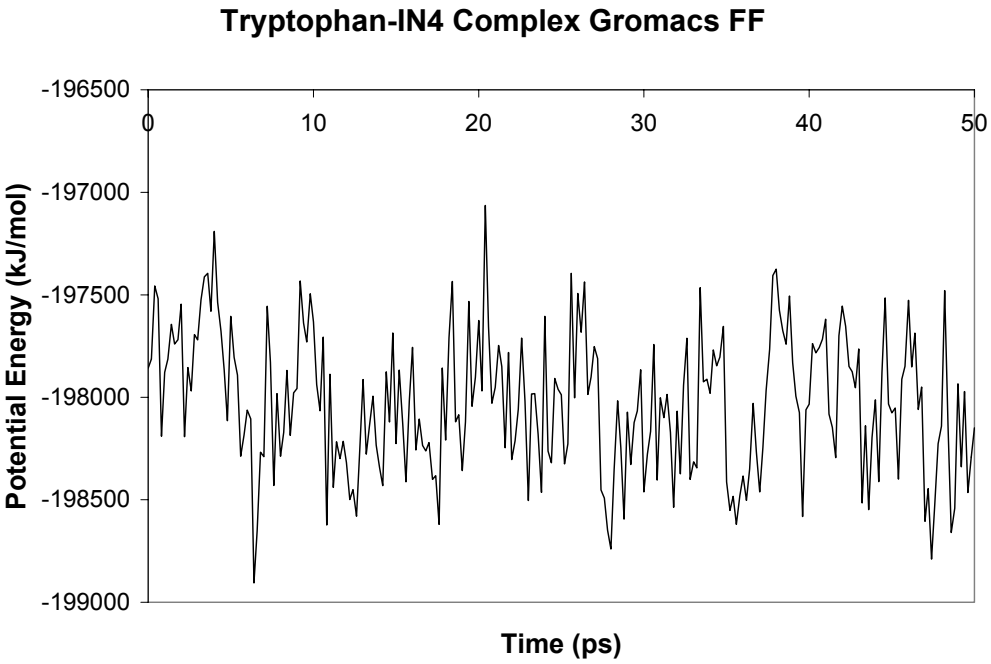
Select the Protein and the DRG groups.

Use **g_energy** to analyze the potential energy output (the md.edr file).

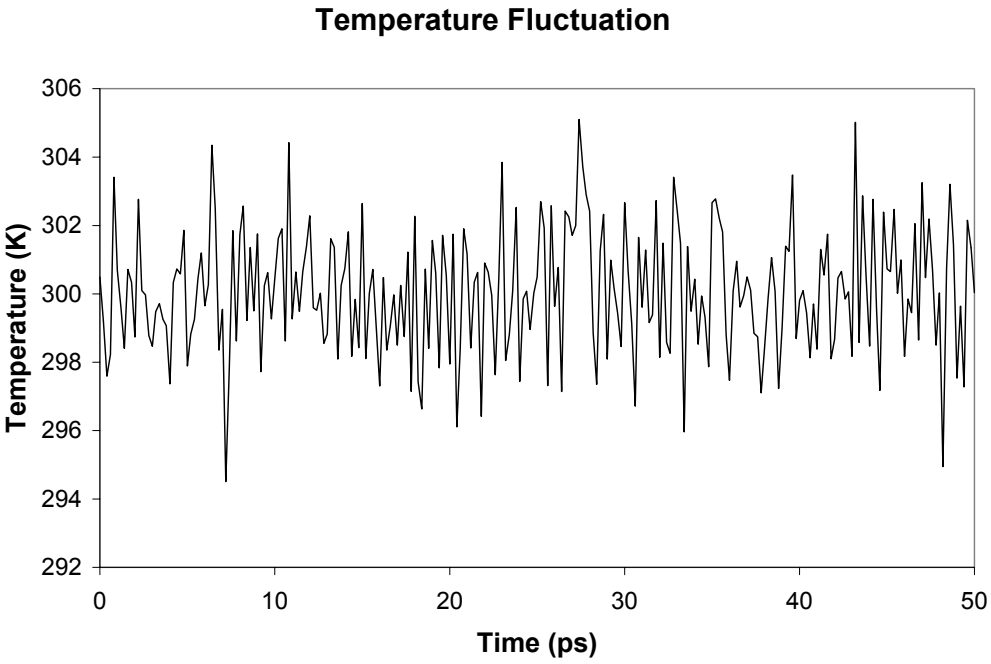
g_energy –f md.edr **–o** pe.xvg

You will be prompted to select the energy term. Enter 10 (for potential) followed by 0 to end selection. Use the g_energy command to obtain other components (e.g. temperature, pressure, kinetic energy, etc.)

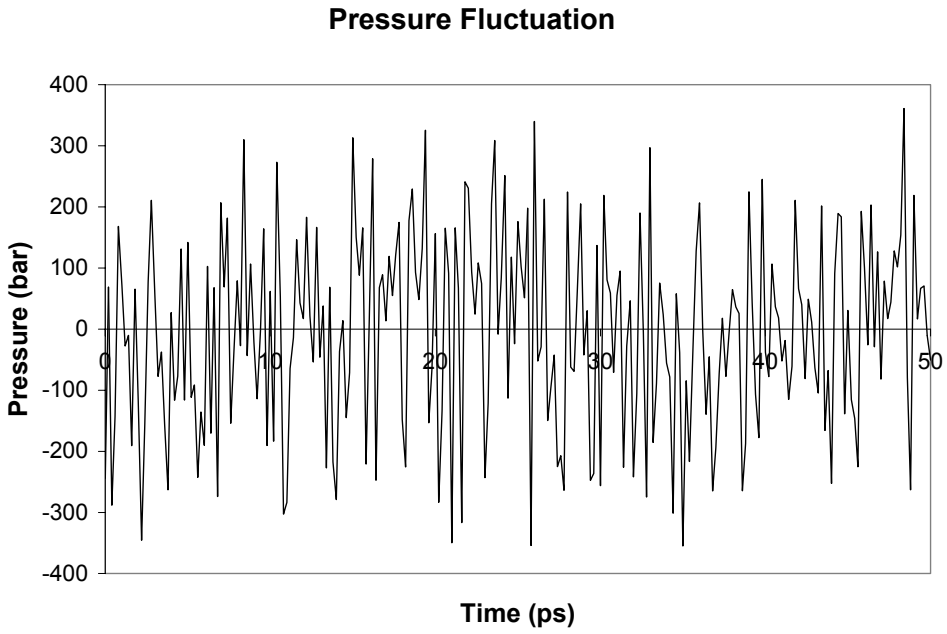
Here is a plot of the potential energy data from our run. We performed a 50 ps dynamics simulation.



The potential energy equilibrates about an average of ~ –198100 kJ/mol.



The temperature plot shows the normal oscillation behavior of the temperature about the desired average (300 K).

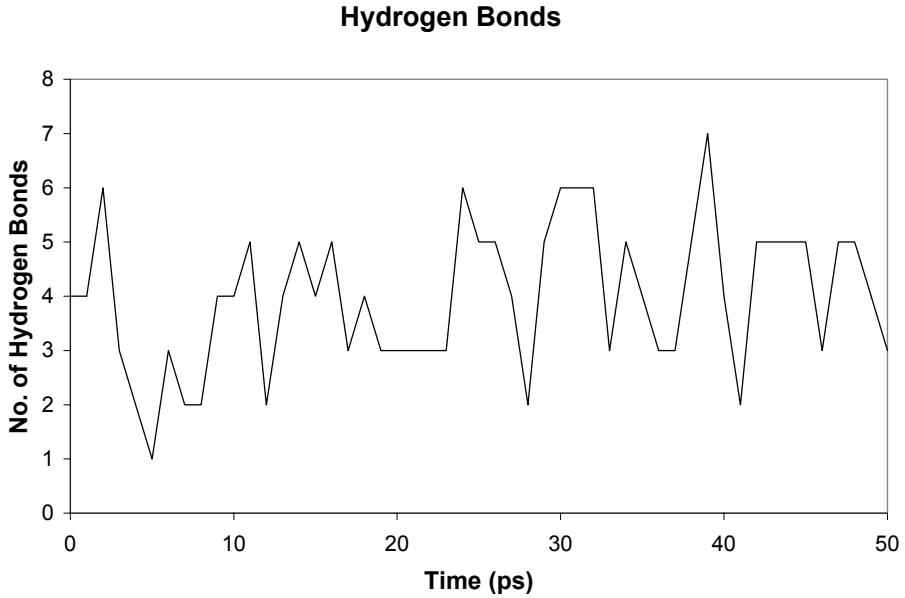


Use **g_hbond** to analyze the number of hydrogen bonds between the drug (DRG) and the enzyme (Protein). Be sure to select the Protein and the DRG groups when prompted by the program.

g_hbond –f trp_md.xtc **–s** trp_md.tpr **–num** trp_hnum.xvg

Select Group 12 (IN4) and Group 1 (Protein).

g_hbond uses a default distance cutoff of 2.5 angstroms (0.25 nm) and a default angle cutoff of 60 degrees (i.e. bond angle must be 60 degrees or greater). The angle cutoff is rather liberal. Some researchers prefer a more stringent angle cutoff of 120 degrees. We used the default in the example above.

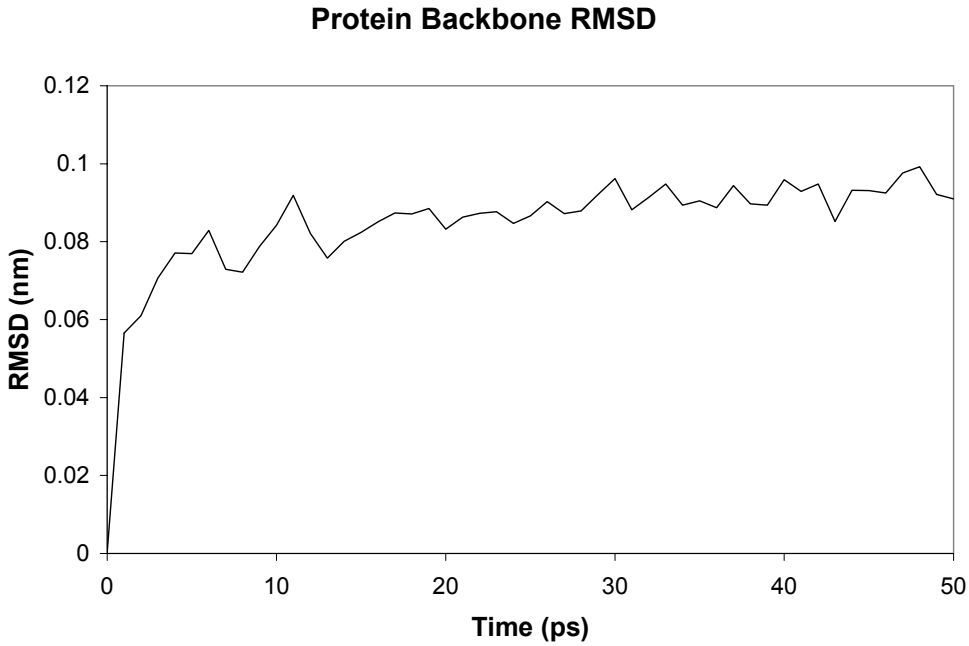


Use **g_rms** to obtain an RMSD plot of the protein backbone and the drug (IN4) throughout the simulation. Do the Backbone first -

g_rms –s trp_md.tpr **–f** trp_md.trr (or xtc) **–o** bkbone_rmsd.xvg

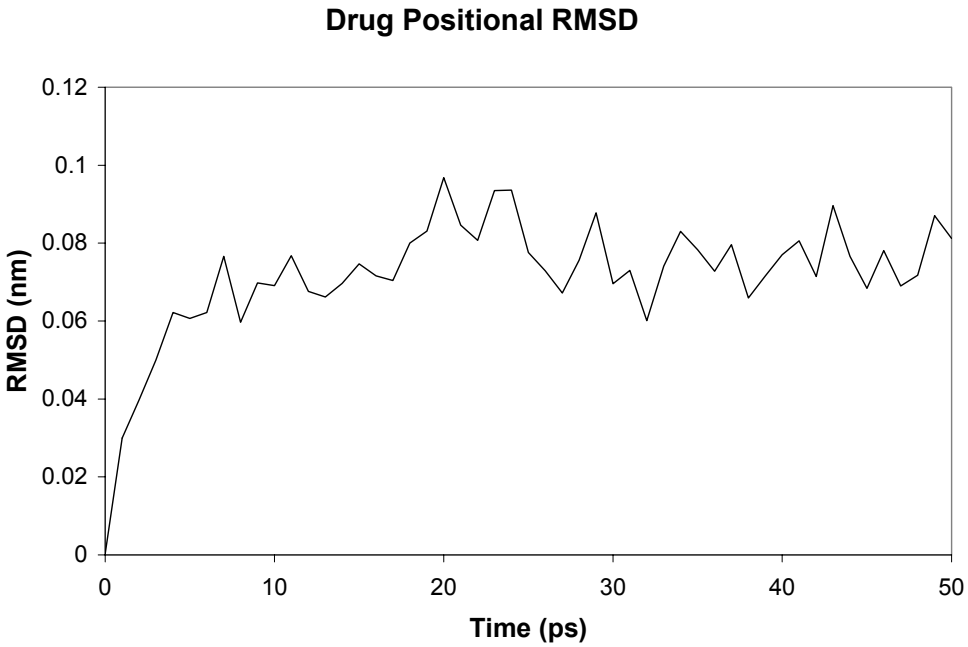
You will be prompted to select a Group. Enter 4 (for backbone).

Compare to 1 Group in the reference. Enter 1 then Enter 4 for backbone again.



Now run the **g_rms** program to obtain an RMS plot for the drug (IN4). Select Group 12 (IN4) in this analysis.

g_rms –s trp_md.tpr **–f** trp_md.trr (or xtc) **–o** drg_rmsd.xvg



The backbone RMSD indicates that the rigid protein structure equilibrates rather quickly in this simulation (after 20 ps). The drug does not equilibrate until after 30 ps. The RMSD for the drug is more variable indicative of its mobility within the binding pocket.

Compute an average structure based upon the equilibration of the drug. We will use the **g_rmsf** program.

g_rmsf –f trp_md.trr **–s** trp_md.tpr **–b** 30 **–e** 50 **–ox** trp_avg.pdb **–o** trp_av.xvg

When prompted to select a Group for the calculation, use the System group (Enter 0).

Refinement of the average structure. Average structures tend to be very crude and require energy minimization. We will perform an in vacuo refinement of our structure. Our first step will require that we

- remove all of the solvent (water) and ions (Cl) in our average structure (use a text editor).
- Separate out the protein (tryspin) into a separate file.
- Run pdb2gmx on the protein.
- Use editconf to convert the drg.pdb to a gro file.
- Append the drg.gro coordinates back to trp.gro as you did before.
- Use the drg.itp file that you generated previously in your top file.
- Run Steepest Descents followed by conjugate gradient using the mdp files that follow.

```
;  Steepest Descents in vacuo
;     User kerrigan (236)
;
;     Input file
;
cpp                 =  /usr/bin/cpp ; location of cpp on linux
constraints         =  none
integrator          =  steep
nsteps              =  400
;
;     Energy minimizing stuff
;
emtol               =  1000
emstep              =  0.01

nstcomm             =  1
ns_type             =  grid
morse               =  no
coulombtype         =  Shift
vdw_type            =  Shift
rlist               =  1.0
rcoulomb            =  1.2
rvdw                =  1.2
rcoulomb_switch     =  1.0
rvdw_switch         =  1.0
epsilon_r           =  6.0
Tcoupl              =  no
Pcoupl              =  no
gen_vel             =  no
```

```
;  Conjugate gradient with morse potential
;   in vacuo
;     User kerrigan (236)
;
;     Input file
;
cpp                 =  /usr/bin/cpp ; location of cpp on linux
constraints         =  none
integrator          =  cg
nsteps              =  3000
;
;     Energy minimizing stuff
;
emtol               =  100
emstep              =  0.01
nstcgsteep          =  1000

nstcomm             =  1
morse               =  yes
coulombtype         =  Shift
vdw_type            =  Shift
ns_type             =  grid
rlist               =  1.0
rcoulomb            =  1.2
rvdw                =  1.2
rcoulomb_switch     =  1.0
```

```
rvdw_switch          =  1.0
epsilon_r            =  6.0
Tcoupl               =  no
Pcoupl               =  no
gen_vel              =  no
```

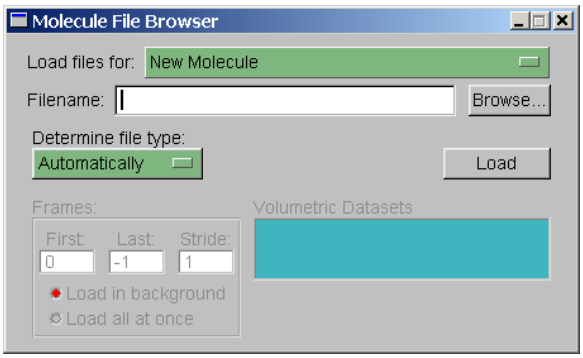
In analyzing a drug/enzyme complex, observations are key. You must ask the following questions:

Was the complex stable to the simulation conditions? (i.e. Did the drug remain in the active site pocket or did it fall out?)

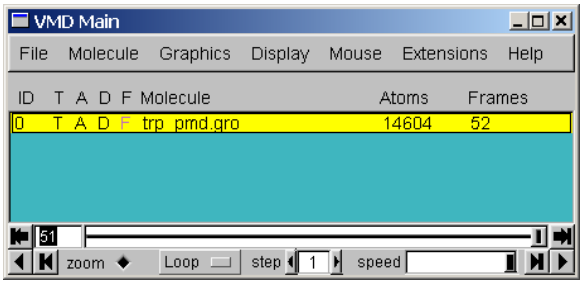
Upon equilibration did the complex become more stable? Why?

What factors contributed to the stability? (hydrogen bonds; hydrophobic pockets; water bridges or other solvent interaction)

Using VMD to view the trajectory. Open VMD. [9] In VMD Main, go to File > New Molecule …



Click on Browse … and select trp_pmd.gro; then, click on Load

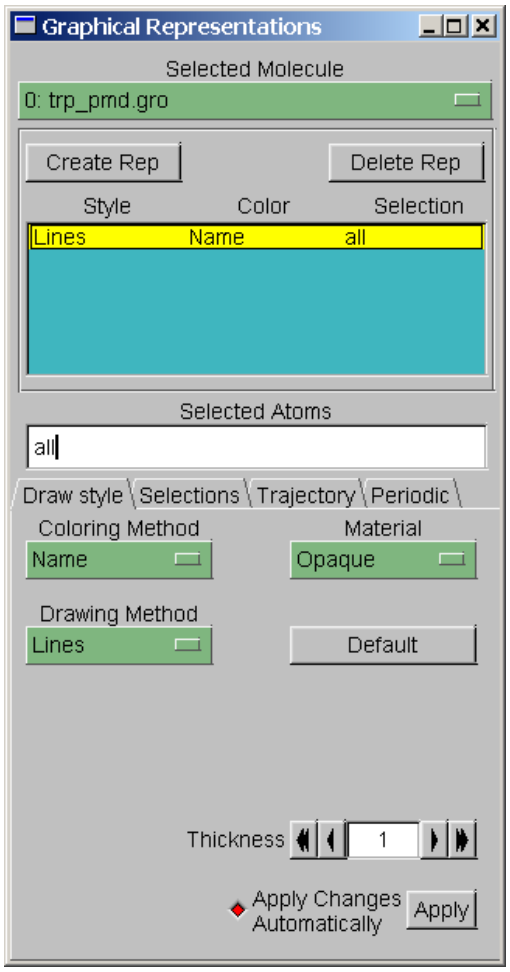


In VMD Main, select File > New Molecule … and select trp_md.xtc (or trr) to load the trajectory. Make sure that the trp_pmd.gro file is highlighted in VMD Main before loading the trajectory.

In VMD Main select Display > Orthographic

Use the mouse to rotate (default mode is rotate). To Zoom or Scale, hit the "s" key to place the mouse in scale mode. Hit the "t" key to use translate mode. Hit the "r" key to go back to rotate mode.

In VMD Main, Open Graphics > Representations

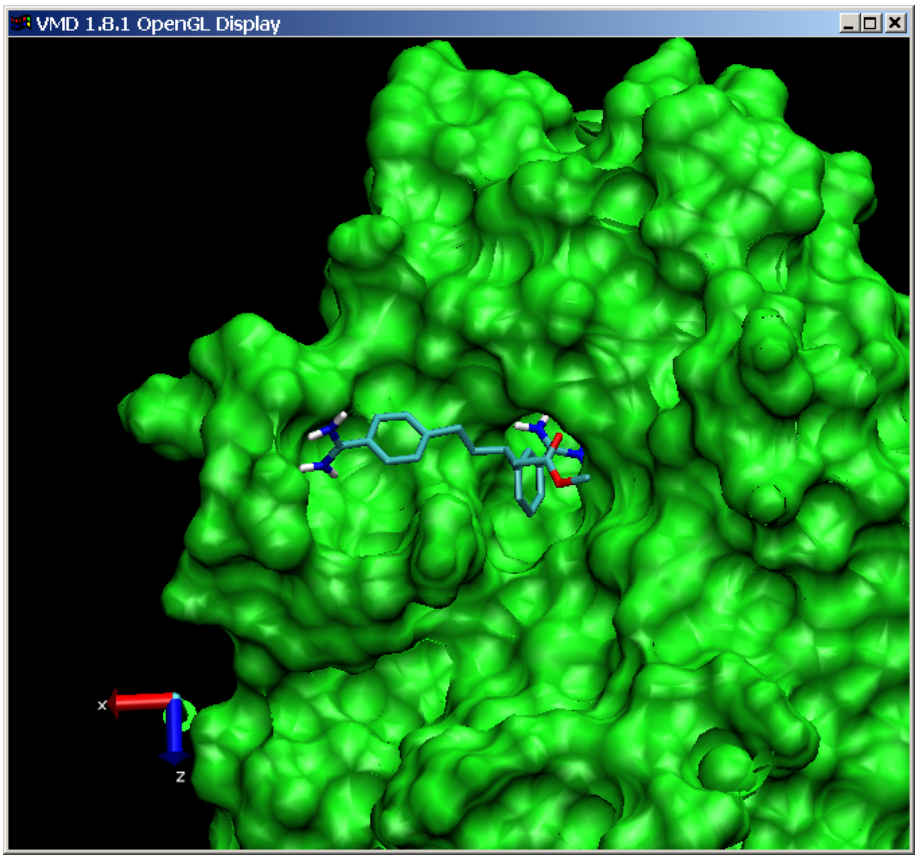


Under selected atoms, type protein in place of all and hit enter (the water goes away and so does your drug). Click on Create Rep to create an additional representation. In the selected atoms box enter resname IN4. Under Drawing Method, select Licorice.

Close the Graphics Representations dialog. In VMD Main, move the speed bar to the midway point and click on the arrow in the lower right-hand corner. This will replay the trajectory over and over again. Note the movement of the drug relative to the protein. Click on the same arrow again to stop the replay. Rotate the structure until you have a nice view of the drug.

Open Graphics Representations again. Select the protein representation. Change Drawing Method to Surf and Coloring Method to Charge. The surface will be colored entirely in green because our input does not have partial atomic charges. Nevertheless, we have a solvent

accessible surface on the receptor illustrating the snug fit of the drug in the binding pocket. VMD uses the Surf program that was developed at UNC-Chapel Hill. [10]



Bibliography

1. Lindahl, E., B. Hess, and D. van der Spoel, *GROMACS 3.0: a package for molecular simulation and trajectory analysis.* J. Mol. Model, 2001. **7**: p. 306-317.
2. van Aalten, D.M.F., R. Bywater, J.B.C. Findlay, M. Hendlich, R.W.W. Hooft, and G. Vriend, *PRODRG, a program for generating molecular toppologies and unique molecular descriptors from coordinates of small molecules.* J. Comput. Aided Mol. Design, 1996. **10**: p. 255-262.
3. Berendsen, H.J., J.P. Postma, W.F. van Gunsteren, and J. Hermans, *Interaction models for water in relation to protein hydration.*, in *Intermolecular Forces*, B. Pullman, Editor. 1981, D. Reidel Publishing Co.: Dordrecht. p. 331-342.
4. Ferguson, D., *Parametrization and evaluation of a flexible water model.* J. Comput. Chem., 1995. **16**: p. 501-511.
5. Darden, T., D. York, and L. Pedersen, *Particle Mesh Ewald: An N-log(N) method for Ewald sums in large systems.* J. Chem. Phys., 1993. **98**: p. 10089-10092.
6. Essmann, U., L. Perera, M.L. Berkowitz, T. Darden, H. Lee, and L. Pedersen, *A smooth particle mesh ewald potential.* J. Chem. Phys., 1995. **103**: p. 8577-8592.
7. Hess, B., H. Bekker, H. Berendsen, and J. Fraaije, *LINCS: A Linear Constraint Solver for molecular simulations.* J. Comp. Chem., 1997. **18**: p. 1463-1472.

8. Berendsen, H.J.C., J.P.M. Postma, W.F. vanGunsteren, A. DiNola, and J.R. Haak, *Molecular dynamics with coupling to an external bath.* J. Chem. Phys., 1984. **81**(8): p. 3584-3590.
9. Humphrey, W., A. Dalke, and K. Schulten, *VMD - Visual Molecular Dynamics.* J. Molec. Graphics, 1996. **14.1**: p. 33-38.
10. Varshney, A., F.P. Brooks, and W.V. Wright, *Linearly Scalable Computation of Smooth Molecular Surfaces.* IEEE Comp. Graphics and Applications, 1994. **14**: p. 19-25.