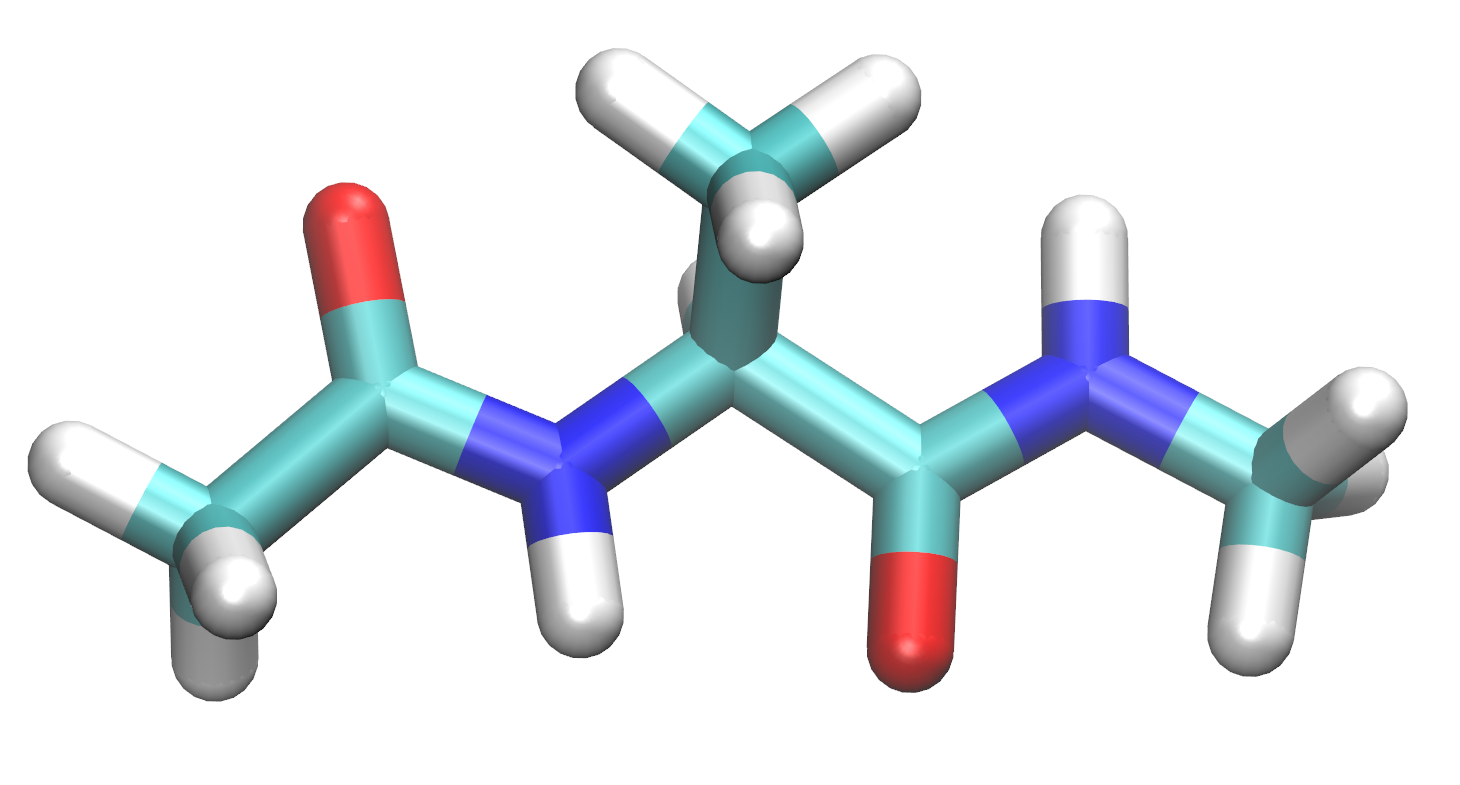
**AMBER Tutorial B0  
  
An Introduction to Molecular Dynamics Simulations using AMBER**

By Benjamin D. Madej & Ross Walker  
  
***Updated for AMBER 15***  
[by Aditi Munshi and Ross Walker]



**TABLE OF CONTENTS**

[Introduction](http://ambermd.org/tutorials/basic/tutorial0/#Introduction)  
[Getting started with Linux](http://ambermd.org/tutorials/basic/tutorial0/#Getting_started_with_Linux_)  
[Prepare topology and coordinate files](http://ambermd.org/tutorials/basic/tutorial0/#Prepare_topology_and_coordinate_files)  
[Load a protein and nucleic acid force field](http://ambermd.org/tutorials/basic/tutorial0/#Load_a_protein_and_nucleic_acid_force_)  
[Build alanine dipeptide](http://ambermd.org/tutorials/basic/tutorial0/#Build_alanine_dipeptide)  
[Solvate alanine dipeptide](http://ambermd.org/tutorials/basic/tutorial0/#Solvate_alanine_dipeptide)  
[Save the Amber prmtop and inpcrd input files](http://ambermd.org/tutorials/basic/tutorial0/#Save_the_Amber_prmtop_and_inpcrd_input_)  
[Prepare Amber MD sander input files](http://ambermd.org/tutorials/basic/tutorial0/#Prepare_Amber_sander_input_files)  
[Minimization input](http://ambermd.org/tutorials/basic/tutorial0/#Minimization_input_)  
[Heating input](http://ambermd.org/tutorials/basic/tutorial0/#Heating_input_)  
[Production input](http://ambermd.org/tutorials/basic/tutorial0/#Production_input_)  
[Run Amber MD sander](http://ambermd.org/tutorials/basic/tutorial0/#Run_Amber_MD_sander_)  
[Run minimization](http://ambermd.org/tutorials/basic/tutorial0/#Run_minimization_)  
[Run heating MD](http://ambermd.org/tutorials/basic/tutorial0/#Run_heating_)  
[Run production MD](http://ambermd.org/tutorials/basic/tutorial0/#Run_production_MD_)  
[Visualize the results](http://ambermd.org/tutorials/basic/tutorial0/#Visualize_the_results_)  
[Analyze the MD results](http://ambermd.org/tutorials/basic/tutorial0/#Analyze_the_MD_results)  
[Production MD system properties](http://ambermd.org/tutorials/basic/tutorial0/#Production_MD_system_properties_)  
[Use cpptraj to analyze the RMSD](http://ambermd.org/tutorials/basic/tutorial0/#Use_cpptraj_to_analyze_the_RMSD_)  
[Conclusion](http://ambermd.org/tutorials/basic/tutorial0/#Conclusion)  
[Appendix: tutorial files](http://ambermd.org/tutorials/basic/tutorial0/#Appendix:_Files)  
[Reference and links](http://ambermd.org/tutorials/basic/tutorial0/#References_and_links)

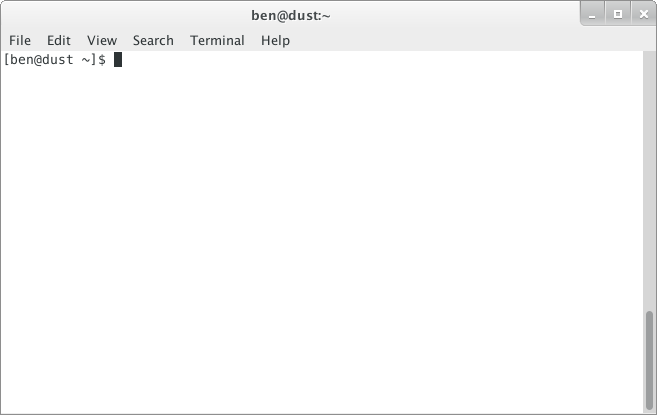
**Introduction**

This tutorial is designed to provide an introduction to molecular dynamics simulations with Amber. It is designed around AMBER Tools v14 and assumes that you have not used Linux or Amber before. It is designed for new users who want to learn about how to run Molecular Dynamics simulations. It does however assume that you have a machine with AmberTools v15, VMD and xmgrace correctly installed.  
  
**AMBER** stands for Assisted Model Building and Energy Refinement. It refers not only to the molecular dynamics programs, but also a set of force fields that describe the potential energy function and parameters of the interactions of biomolecules.  
  
In order to run a Molecular Dynamics simulation in Amber, each molecule's interactions are described by a molecular force field. The force field has specific parameters defined for each molecule.  
  
**sander** is the basic MD engine of Amber. **pmemd** is the high performance implementation of the MD engine that contains a subset of features of sander. **pmemd** can also be run with acceleration from graphics processing units (GPU).  
  
In order to run an MD simulation with sander or pmemd, three key files are needed:

1. **prmtop** - The file that describes the parameter and topology of the molecules in the system
2. **inpcrd** - The file that describes the initial molecular coordinates of the system
3. **mdin** - The file that describes the settings for the Amber MD engine

**Getting started with Linux**

Amber MD is software that is entirely based on a Command Line Interface (CLI) on a computer with Linux. To run Amber, you will need to open a terminal.  
  
1. Open a terminal now on your Linux computer.  
  
On most Linux machines, your terminal will look like this:



Most of the work for this tutorial will be done mainly through the terminal.

**List files and make a directory (folder) to store your files in**

When you first log in and start a terminal, your current working directory (or folder) is your home directory. It has the same name as your username, and it is where your files and directories are stored. In most cases, this is **/home/username**

**ls (list)**

2. Use the **ls** command to list what is in your current directory.  
Note that the "$" is the command line prompt in the terminal.

$ ls

At this point there will probably be some files and directories in your home directory that are created automatically with your account.

**mkdir (make directory)**

You'll need a new directory for the files and folders created in this tutorial.

3. Make a new directory for this tutorial with the **mkdir** command called **Tutorial**.

$ mkdir Tutorial

Now when you do **ls** you should see your new directory has been created.

$ ls  
**Tutorial**

**cd (change directory)**

At this point, you'll want to move into your Tutorial directory so that you can save all of your working files there.

4. Use the **cd** command to change to different directory.

$ cd Tutorial  
$ ls

There is a special directory named "..". This means the parent of the current directory. So to return to the parent of the Tutorial directory use **cd ..**

$ cd ..  
$ ls  
**Tutorial**

If you ever need to return to your home directory use just the **cd** command by itself. Tilde "**~**" is a shortcut to your home directory. The following commands both change directory to your home directory.

$ cd  
$ cd ~

**pwd (print working directory)**

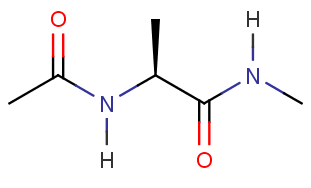
Pathnames describe what directory you are in relative to the entire computer's filesystem. You home directory has location within the entire filesystem.  
  
5. Print the working directory pathname of your home directory with **pwd**.

$ cd  
$ pwd  
/home/username

This is the current working directory that you're located in. In this case the directory **username** is in the directory **home** which is in the **/**(root) directory.

**Prepare topology and coordinate files**

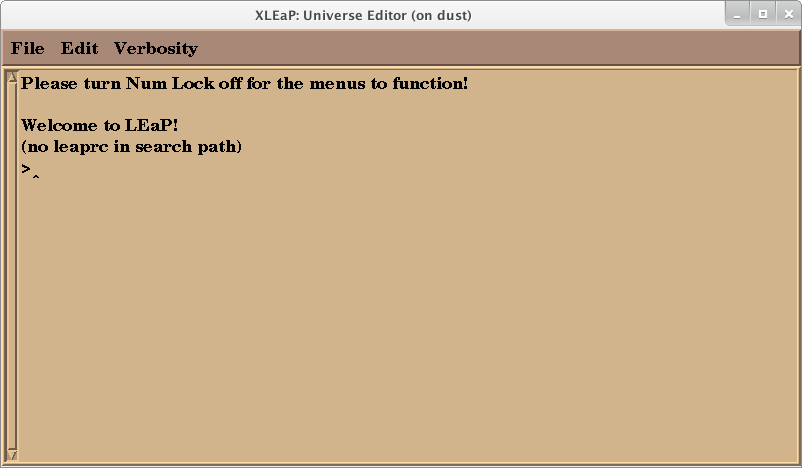
For this tutorial, you will build the following molecule in the preparatory program called xLEaP for simulation in AMBER.



In order to build and solvate this molecule, you will need to start xLEaP. xLEaP has another command line interface and simple molecular graphics for building the system topology and define parameters for the molecules.  
  
6. Start xLEaP now with the **xleap** command.

$ xleap  
-I: Adding /usr/local/amber\_14/amber/dat/leap/prep to search path.  
-I: Adding /usr/local/amber\_14/amber/dat/leap/lib to search path.  
-I: Adding /usr/local/amber\_14/amber/dat/leap/parm to search path.  
-I: Adding /usr/local/amber\_14/amber/dat/leap/cmd to search path.

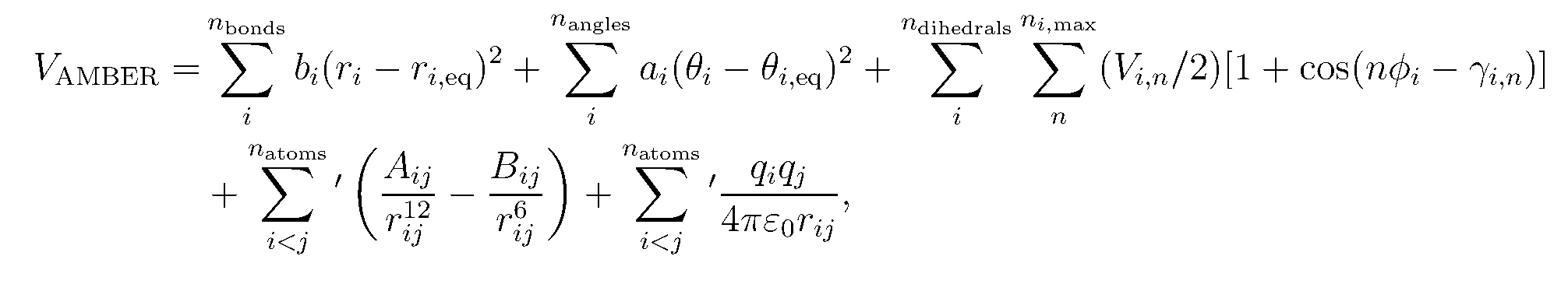
You should see a window like this:



Warning:  
Do NOT click the "X" on any LEaP window. It will quit LEaP entirely.  
  
Note:  
At this point it's a good idea to turn Num Lock off for the menus to work.

**Load a protein and nucleic acid force field**

A MD force field is the the Hamiltonian (potential energy function) and the related parameters that describe the intra- and intermolecular interactions between the molecules in the system. In MD, the Hamiltonian is integrated to describe the forces and velocities of the molecules.  
  
The basic form of the Amber Hamiltonian is:



In order to run a molecular dynamics simulation, we need to load a force field to describe the potential energy of alanine dipeptide. We will use the AMBER force field FF14SB for proteins and nucleic acids. FF14SB is based off FF12SB, an updated version of FF99SB, which in turn was based on the original Amber Cornell *et al* (1995) [ff94] force field. The most significant changes to the FF14SB force field included updated torsion terms for the protein Phi-Psi angles and refits of the torsion terms for side chains. Together these improved the estimation of alpha helices in these molecules.  
  
7. **source** (load) the FF14SB force field now.

> source leaprc.ff14SB  
Loading parameters:  
/usr/local/amber14/dat/leap/parm/frcmod.ff14SB  
Reading force field modification type file (frcmod)  
Reading title:

**Build alanine dipeptide**

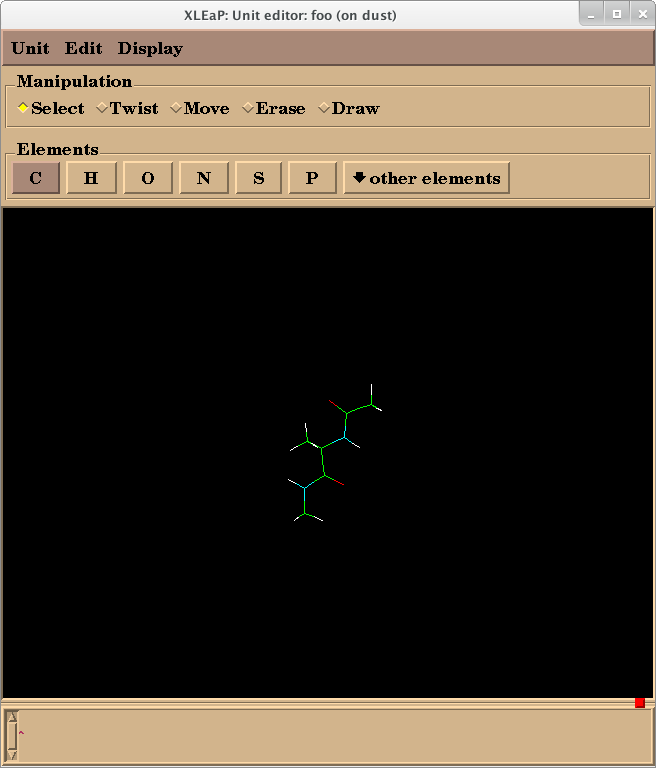
We can build an alanine dipeptide as an alanine amino acid capped with an acetyl group on the N-terminus and n-methylamide on the C-terminus. After we loaded the force field ff14SB, xLEaP now has these "units" available to build into a molecule. The sequence command will create a new unit from the available units and connect them together.  
  
8. Use **sequence** to create a new unit called **foo** out of the **ACE**, **ALA**, and **NME** units.  
Note that the ">" indicates the command line in xLEaP.

> foo = sequence { ACE ALA NME }

Now you have a single alanine dipeptide molecule stored in the unit **foo**. xLEaP provides a very basic editor to examine and change units and molecules.  
  
9. Examine the structure of the alanine dipeptide molecule. Use the **edit** command to view the structure.

> edit foo

The editing window will look like this:



From here, you can examine the topology, structure, atom names, atom types, and partial charges of the molecule. Basic editing of the molecule is also possible.  
  
Warning:  
Do NOT click the "X" to close this window. It will exit xLEaP. To close this window, use **Unit -> Close**.

**Solvate alanine dipeptide**

The next step to prepare this alanine dipeptide system is to solvate the molecule with explicit water molecules. In this simulation we will use add TIP3P water molecules to the system.

In this type of simulation, the system has periodic boundary conditions, meaning that molecules that exit one side of the system will wrap to the other side of the system. It is important that the periodic box is large enough, i.e. there is enough water surrounding alanine dipeptide, so that the alanine dipeptide molecule does not interact with its periodic images.

There are many water models available for MD simulations. However, for this tutorial we will use the TIP3P water model for this simulation.

10. Solvate the system with the **solvatebox** command.

> solvatebox foo TIP3PBOX 10.0

**TIP3PBOX** specifies the type of water box to solvate with. **10.0** indicates that the molecule should have a buffer of at least 10 Angstroms between alanine dipeptide and the periodic box wall.

**Save the Amber prmtop and inpcrd input files**

Now we will save the **prmtop** and **inpcrd** files to the current working directory. The unit **foo** now contains the alanine dipeptide molecule, water molecules, and the periodic box information necessary for simulation. The parameters will be assigned from the ff99SB force field.

11. To save the prmtop and inpcrd file use the **saveamberparm** command.

> saveamberparm foo prmtop inpcrd  
Checking Unit.  
Building topology.  
Building atom parameters.  
Building bond parameters.  
Building angle parameters.  
Building proper torsion parameters.  
Building improper torsion parameters.  
 total 4 improper torsions applied  
Building H-Bond parameters.  
Incorporating Non-Bonded adjustments.  
Not Marking per-residue atom chain types.  
Marking per-residue atom chain types.  
 (Residues lacking connect0/connect1 -  
 these don't have chain types marked:  
  
 res total affected  
  
 WAT 630  
 )  
 (no restraints)

Warning:  
Pay close attention to the output of this command for any warnings or errors that your **prmtop** and **inpcrd**file did not build correctly.

**Amber parameter/topology and coordinate files**

The alanine dipeptide **prmtop** and **input** files are available here:  
[prmtop](http://ambermd.org/tutorials/basic/tutorial0/include/prmtop)  
[inpcrd](http://ambermd.org/tutorials/basic/tutorial0/include/inpcrd)

**Quit xLEaP**

12. To quit xLEaP use **quit.**

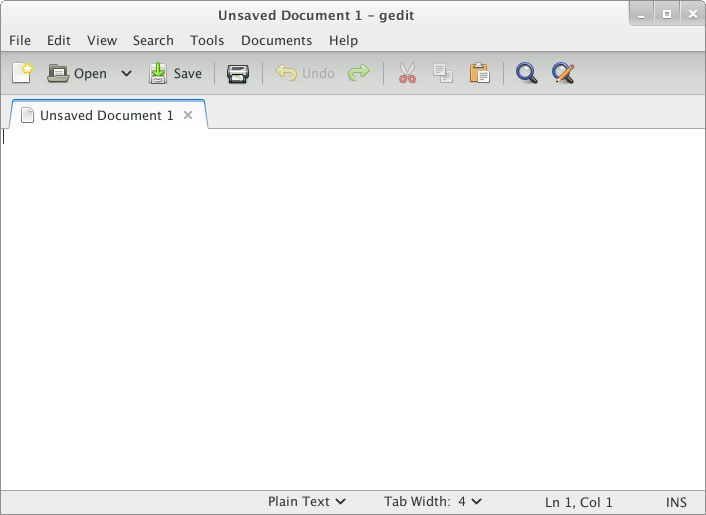
> quit

**Prepare Amber MD sander input files**

The last components needed are the input files that define the program settings for each MD run. For this system, we will perform an energy minimization on the system, then slowly heat the system, and then do production MD at the desired temperature and pressure.

1. Minimization
2. Heating with constant volume and temperature (NVT) for 20ps from 0K to 300K
3. Production MD with constant pressure and temperature (NPT) at 300K and 1atm for 60ps

We will save the trajectory and write to the output file every 2ps. The Langevin thermostat will be used to control the temperature. The random number generator will be initialized with a random seed.  
  
To control all these settings, we will write a simple input file in a text editor. Linux has many text editors available, but we will use a simple text editor included on your linux computer.  
  
13. Open the **gedit Text Editor** on your Linux computer.  
  
gedit's interface looks like this:



**Minimization input**

14. Create the file **01\_Min.in** that includes the following settings for minimization:

Minimize  
 &cntrl  
 imin=1,  
 ntx=1,  
 irest=0,  
 maxcyc=2000,  
 ncyc=1000,  
 ntpr=100,  
 ntwx=0,  
 cut=8.0,  
 /

The settings can be summarized as follows:

|  |  |
| --- | --- |
| imin=1 | Choose a minimization run |
| ntx=1 | Read coordinates but not velocities from ASCII formatted **inpcrd**coordinate file |
| irest=0 | Do not restart simulation. (not applicable to minimization) |
| maxcyc=2000 | Maximum minimization cycles |
| ncyc=1000 | The steepest descent algorithm for the first 0-**ncyc** cycles, then switches the conjugate gradient algorithm for **ncyc**-**maxcyc**cycles |
| ntpr=100 | Print to the Amber **mdout** output file every **ntpr** cycles |
| ntwx=0 | No Amber **mdcrd**trajectory file written (not applicable to minimization) |
| cut=8.0 | Nonbonded cutoff distance in Angstroms (for PME, limit of the direct space sum - do NOT reduce this below 8.0. Higher numbers give slightly better accuracy but at vastly increased computational cost.) |

**Heating input**

15. Create the file **02\_Heat.in** that includes the following settings for heating:

Heat  
 &cntrl  
 imin=0,  
 ntx=1,  
 irest=0,  
 nstlim=10000,  
 dt=0.002,  
 ntf=2,  
 ntc=2,  
 tempi=0.0,  
 temp0=300.0,  
 ntpr=100,  
 ntwx=100,  
 cut=8.0,  
 ntb=1,  
 ntp=0,  
 ntt=3,  
 gamma\_ln=2.0,  
 nmropt=1,  
 ig=-1,  
 /  
&wt type='TEMP0', istep1=0, istep2=9000, value1=0.0, value2=300.0 /  
&wt type='TEMP0', istep1=9001, istep2=10000, value1=300.0, value2=300.0 /  
&wt type='END' /

The settings can be summarized as follows:

|  |  |
| --- | --- |
| imin=0 | Choose a molecular dynamics (MD) run [no minimization] |
| nstlim=10000 | Number of MD steps in run (nstlim \* dt = run length in ps) |
| dt=0.002 | Time step in picoseconds (ps). The time length of each MD step |
| ntf=2 | Setting to not calculate force for SHAKE constrained bonds |
| ntc=2 | Enable SHAKE to constrain all bonds involving hydrogen |
| tempi=0.0 | Initial thermostat temperature in K (see **NMROPT** section) |
| temp0=300.0 | Final thermostat temperature in K (see **NMROPT** section) |
| ntwx=1000 | Write Amber trajectory file **mdcrd** every **ntwx** steps |
| ntb=1 | Periodic boundaries for constant volume |
| ntp=0 | No pressure control |
| ntt=3 | Temperature control with Langevin thermostat |
| gamma\_ln=2.0 | Langevin thermostat collision frequency |
| nmropt=1 | NMR restraints and weight changes read (see **NMROPT** section) |
| ig=-1 | Randomize the seed for the pseudo-random number generator [always a good idea unless you are debugging a simulation problem] |

**Thermostat temperature via NMROPT**

The final three lines allow the thermostat to change its target temperature throughout the simulation. For the first 9000 steps, the temperature will increase from 0K to 300K. For steps 9001 to 10000, the temperature will remain at 300K.

**Production input**

Warning:  
**By itself, this input file is not intended for general MD simulations.**  
  
**NTPR and NTWX are set very low** so that it is possible to analyze this short simulation. Using these settings for longer MD simulations will create very large output and trajectory files and will be slower than regular MD. **For real production MD, you'll need to increase NTPR and NTWX.**  
  
The production time of this simulation is only 60ps. Ideally, we would run this simulation for much longer, but in the interest of time for this tutorial, we have limited the production simulation time.  
  
16. Create the file **03\_Prod.in** with the settings for production MD:

Production  
 &cntrl  
 imin=0,  
 ntx=5,  
 irest=1,  
 nstlim=30000,  
 dt=0.002,  
 ntf=2,  
 ntc=2,  
 temp0=300.0,  
 ntpr=100,  
 ntwx=100,  
 cut=8.0,  
 ntb=2,  
 ntp=1,  
 ntt=3,  
 gamma\_ln=2.0,  
 ig=-1,  
 /

The settings for production can be summarized as follows:

|  |  |
| --- | --- |
| ntx=5 | Read coordinates and velocities from unformatted **inpcrd**coordinate file |
| irest=1 | Restart previous MD run [This means velocities are expected in the inpcrd file and will be used to provide initial atom velocities] |
| temp0=300.0 | Thermostat temperature. Run at 300K |
| ntb=2 | Use periodic boundary conditions with constant pressure |
| ntp=1 | Use the Berendsen barostat for constant pressure simulation |

**Input files**

The **sander** input files are available here:  
[01\_Min.in](http://ambermd.org/tutorials/basic/tutorial0/include/01_Min.in)  
[02\_Heat.in](http://ambermd.org/tutorials/basic/tutorial0/include/02_Heat.in)  
[03\_Prod.in](http://ambermd.org/tutorials/basic/tutorial0/include/03_Prod.in)

**Run Amber MD sander**

Now that we have all the ingredients: the parameter and topology file **prmtop**, the coordinate file **inpcrd**, and the input files **01\_Min.in**, **02\_Heat.in**, **03\_Prod.in**, we are ready to run the actual minimization, heating, and production MD.  
  
To do this, we will use the program **sander**, the general purpose MD engine of Amber (there is also a high performance version, termed **pmemd**, which is part of the commercial version of AMBER and the optimum choice of MD engine but for purposes of the tutorial **sander** will suffice). **sander** is run from the command line. On the command line, we can specify several more options and choose which files are to be used for input.  
  
17. First, from the terminal, you will need to change directories to the **Tutorial** directory with all of your input files. "**~**" is a shortcut to your home directory where you created a **Tutorial** directory.

$ cd ~/Tutorial

**Run minimization**

18. Run the minimization of alanine dipeptide with **sander.**

$ $AMBERHOME/bin/sander -O -i 01\_Min.in -o 01\_Min.out -p prmtop -c inpcrd -r 01\_Min.rst \

-inf 01\_Min.mdinfo

sander uses a consistent syntax for each step of MD simulation. Here is a summary of the command line options of **sander**:

|  |  |
| --- | --- |
| -O | Overwrite the output files if they already exist |
| -i 01\_Min.in | Choose input file (default **mdin**) |
| -o 01\_Min.out | Write output file (default **mdout**) |
| -p prmtop | Choose parameter and topology file **prmtop** |
| -c inpcrd | Choose coordinate file **inpcrd** |
| -r 01\_Min.rst | Write output restart file with coordinates and velocities (default **restrt**) |
| -inf 01\_Min.mdinfo | Write MD info file with simulation status (default **mdinfo**) |

**sander** should complete the minimization in a moderate amount of time (~ 27 seconds) depending on your computer specifications.  
  
After sander completes, there should be an output file **01\_Min.out**, a restart file **01\_Min.rst**, and a MD info file **01\_Min.mdinfo**. You will use the restart file **01\_Min.rst** for the heating of the system.

**Minimization output files**

The minimization output files are available here:  
[01\_Min.out](http://ambermd.org/tutorials/basic/tutorial0/include/01_Min.out)  
[01\_Min.rst](http://ambermd.org/tutorials/basic/tutorial0/include/01_Min.rst)

19. Using gedit, open the output file **01\_Min.out**.  
  
In the **01\_Min.out** file, you will find the details of your minimization. You should be able to see the system energy **ENERGY** decrease throughout the minimization.

**Run heating MD**

Now, using the restart file from the initial minimization, we will heat the system.

20. Run the heating of alanine dipeptide with **sander**.

$ $AMBERHOME/bin/sander -O -i 02\_Heat.in -o 02\_Heat.out -p prmtop -c 01\_Min.rst \

-r 02\_Heat.rst -x 02\_Heat.mdcrd -inf 02\_Heat.mdinfo

Here is a summary of the command line options for **sander:**

|  |  |
| --- | --- |
| -c 01\_Min.rst | Now for the input coordinates we choose the restart file from minimization |
| -x 02\_Heat.mdcrd | Output trajectory file for MD simulation (default **mdcrd**) |

**sander** should complete the heating in a moderate amount of time (~ 2.5 mins) depending on your computer specifications.

**Heating output files**

The heating output files are available here. Some files are compressed and need to be unzipped.  
[02\_Heat.out](http://ambermd.org/tutorials/basic/tutorial0/include/02_Heat.out)  
[02\_Heat.rst](http://ambermd.org/tutorials/basic/tutorial0/include/02_Heat.rst)  
[02\_Heat.mdcrd](http://ambermd.org/tutorials/basic/tutorial0/include/02_Heat.mdcrd.gz)  
  
21. Open the output file **02\_Heat.out** to view the system output.  
  
In the **02\_Heat.out** file you will find the output from the heating MD. You should be able to see system information including timestep energies, and temperature. For example on the 1000 time step:

NSTEP = 1000 TIME(PS) = 2.000 TEMP(K) = 29.48 PRESS = 0.0  
 Etot = -6944.9552 EKtot = 112.3015 EPtot = -7057.2567  
 BOND = 1.0442 ANGLE = 1.7653 DIHED = 9.4906  
 1-4 NB = 2.6284 1-4 EEL = 46.3073 VDWAALS = 1448.7074  
 EELEC = -8567.1999 EHBOND = 0.0000 RESTRAINT = 0.0000  
 Ewald error estimate: 0.4641E-03  
 ------------------------------------------------------------------------------  
  
 NMR restraints: Bond = 0.000 Angle = 0.000 Torsion = 0.000  
===============================================================================

Some of the important values include:

|  |  |
| --- | --- |
| NSTEP | The time step that the MD simulation is at |
| TIME | The total time of the simulation (including restarts) |
| TEMP | System temperature |
| PRESS | System pressure |
| Etot | Total energy of the system |
| EKtot | Total kinetic energy of the system |
| EPtot | Total potential energy of the system |

Note that the pressure is 0.0 because the barostat (pressure control) is not being used in the heating.

**Run production MD**

Now that minimization and heating are complete. We move on to the actual production MD.  
  
22. Run the production MD of alanine dipeptide with **sander**.

$ $AMBERHOME/bin/sander -O -i 03\_Prod.in -o 03\_Prod.out -p prmtop -c 02\_Heat.rst \

-r 03\_Prod.rst -x 03\_Prod.mdcrd -inf 03\_Prod.info &

Note: With the "&" at the end of the command, **sander** now runs in the background  
  
Now **sander** is running in the background. It will take some time to run the production MD.   
  
But we'd like to monitor the status of the production MD. So we will monitor the sander output file to check the status of the run. The Linux program **tail** will print the end of the output file.  
  
23. To monitor the status of this job use the program **tail** to print the output file to the terminal.

$ tail -f 03\_Prod.out

This prints the output file as **sander** is running. It's useful to keep track of the job. You can also monitor the mdinfo file (cat 03\_Prod.info) which provides detailed performance data as well as estimated time to completion.  
  
24. To exit **tail**, quit the program by pressing <CTRL-C>.

**Complete the MD simulation**

Let the MD simulation run. It should take a while to complete the simulation [~ 10 minutes].

**Production output**

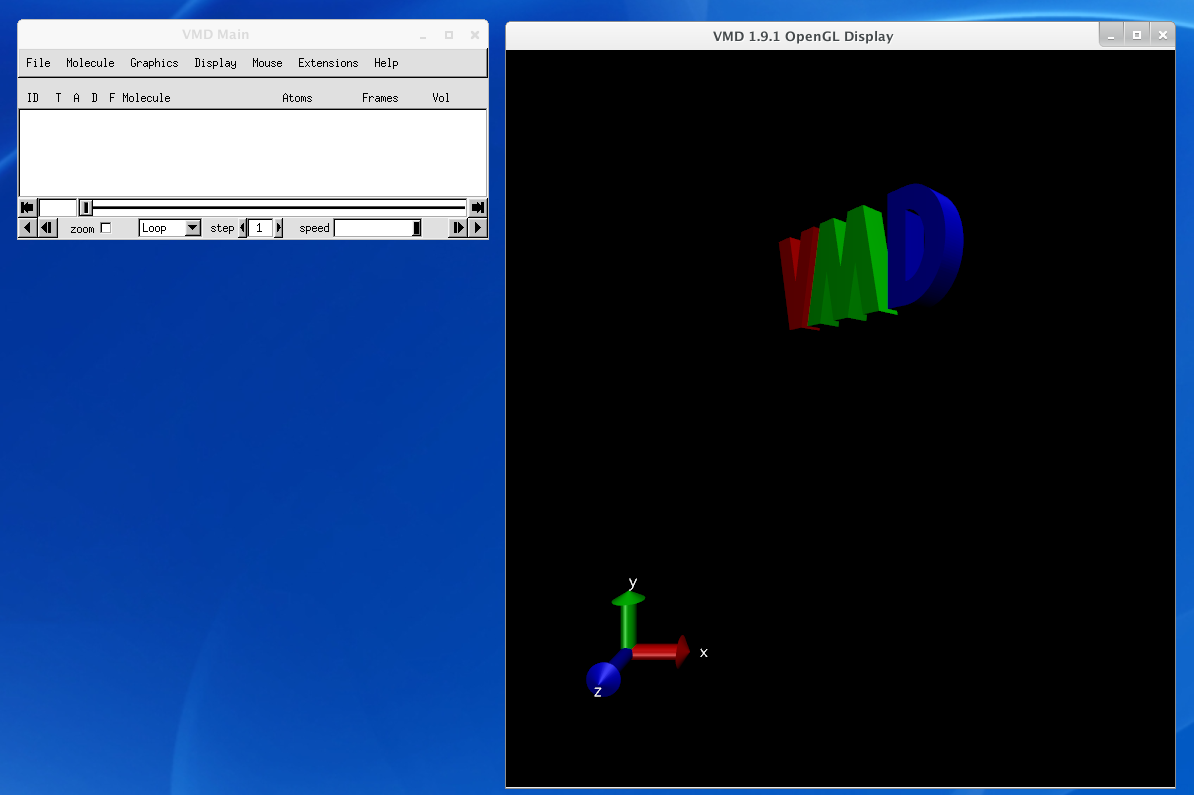
The production MD output is available here:  
[03\_Prod.out](http://ambermd.org/tutorials/basic/tutorial0/include/03_Prod.out)  
[03\_Prod.rst](http://ambermd.org/tutorials/basic/tutorial0/include/03_Prod.rst)  
[03\_Prod.mdcrd](http://ambermd.org/tutorials/basic/tutorial0/include/03_Prod.mdcrd.gz)  
  
Once it's complete open the output file to check that the simulation completed normally.  
  
25. Open the output file **03\_Prod.out** with gedit to view the output of the MD simulation.

**Visualize the results**

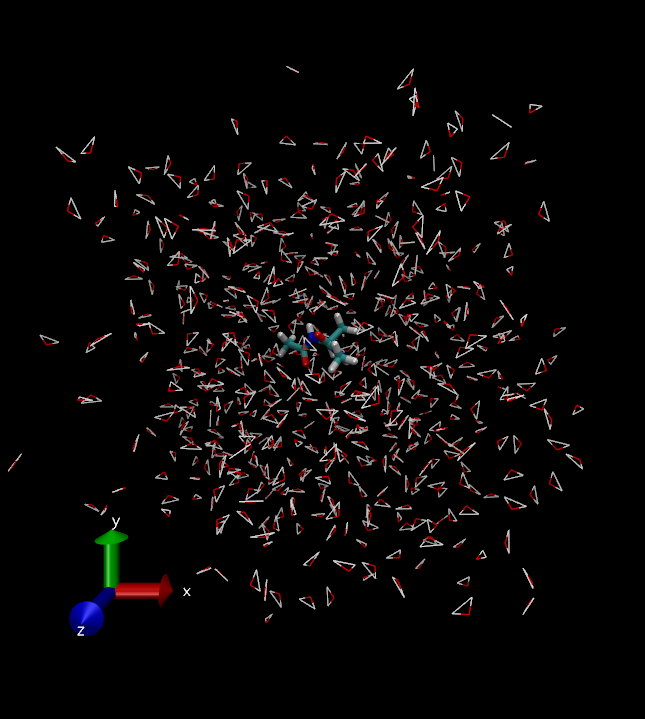
You've now run an MD simulation. In order to visualize the results, we will now use a program called VMD (Visual Molecular Dynamics). This is a molecular graphics program that can render molecular structures in 3D. VMD not only loads Protein Database (PDB) structure files, but also MD trajectories from many programs. [A more in-depth tutorial on VMD is available as an optional hands-on session listed at the bottom of the workshop program page].  
  
26. To start VMD, open a terminal, change directory to your tutorial files in the Tutorial directory, and run**vmd**. Remember **~/Tutorial** is a shortcut to your **Tutorial** directory.

$ cd ~/Tutorial  
$ vmd

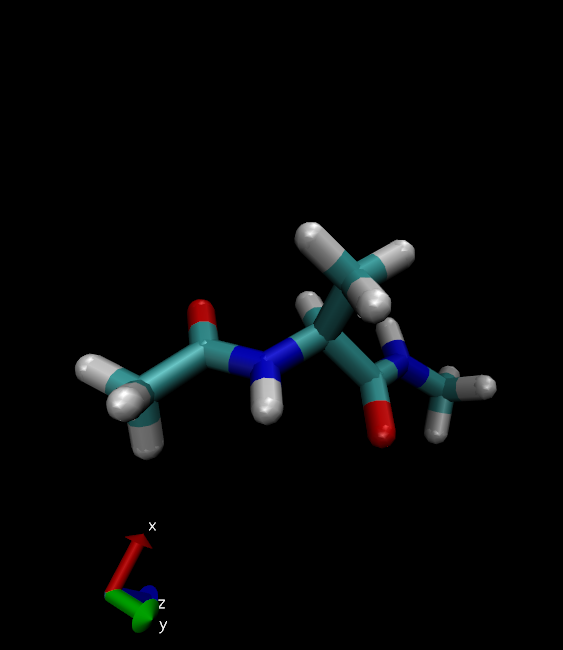
VMD should look like this:



VMD is a very useful tool to visualize protein, nucleic acid, and other biomolecular atomic structures. One of the most common formats is the PDB biomolecular structure format. To load a PDB, got to **File -> New Molecule...**. Then load files for a **New Molecule** and choose the PDB file. VMD should determine the file type**Automatically**.  
  
However, we would like to visualize the alanine dipeptide trajectory. Now we will load our MD trajectory to look at the dynamics of alanine dipeptide.  
  
27. Create a new molecule with **File -> New Molecule...**  
  
28. Load files for **New Molecule**. Then choose the Amber parameter and topology file **prmtop**. Set the file type to **AMBER7 Parm**. Click **Load**.  
  
29. Load files for **0: prmtop**. Then choose the Amber trajectory file **03\_Prod.mdcrd**. Set the file type to**AMBER Coordinates with Periodic Box**. Click **Load**.  
  
VMD now loads your trajectory to be visualized. The main VMD window can be used to control playback.   
  
You should be able to see the alanine dipeptide molecule as well as the many water molecules in the display. You can rotate, zoom and pan the molecules around the display with the mouse.



Many different visualization options can be changed in the **Graphics -> Representations** window.  
  
Your visualization can be restricted to the alanine dipeptide as well.  
  
30. Change the **Selected Atoms** to **all not water**.  
  
You can change the drawing method for the molecule to a more interesting model.  
  
31. Change the **Drawing Method** to **Licorice**.  
  
Alanine dipeptide will look something like this:



**More VMD**

VMD has a lot of functions that can be used to analyze and study a MD trajectory. For example it is possible to align molecules, measure root mean squared deviations (RMSD), save structures from a trajectory, and measure physical system parameters throughout a trajectory. It's also possible to render a movie of a trajectory.  
  
However, these functions are beyond the scope of this introductory section. For more details refer to the VMD tutorial on the main AMBER Tutorial page.

**Analyze the MD results**

Amber includes a suite of tools to examine and analyze MD trajectories. In this tutorial, we will do a simple analysis with several Amber programs and plot the results. The analysis will primarily done from the command line in the terminal.  
  
32. Open a terminal and change directory to your tutorial files.

$ cd ~/Tutorial

33. Make an **Analysis** directory and change to that directory.

$ mkdir Analysis  
$ cd Analysis

Now we will use an analysis script **process\_mdout.perl** to analyze the MD output files. This script will extract the energies, temperature, pressure, density, and volume from the MD output files and save them to individual data files.  
  
34. Process the MD output files with **process\_mdout.perl**

$ $AMBERHOME/bin/process\_mdout.perl ../02\_Heat.out ../03\_Prod.out

It is now quite simple to plot the data saved in the output files.  
  
We will use a convenient, simple plotting program called **xmgrace** to automatically generate plots for the following MD simulation properties throughout the simulation. We use this for our convenvience, but you can use any plotting program of your choice.

1. MD simulation temperature
2. MD simulation density
3. MD simulation total, potential, and kinetic energies.

However, for the MD simulation density, the heating portion of the simulation does not include a density output. You will need to edit the **summary.DENSITY** file to remove the empty data points for **xmgrace** to work.  
  
35. Use **gedit** to edit the **summary.DENSITY** file to remove the empty data points (up to 20ps).  
  
36. Plot these properties through the simulation using the following commands.

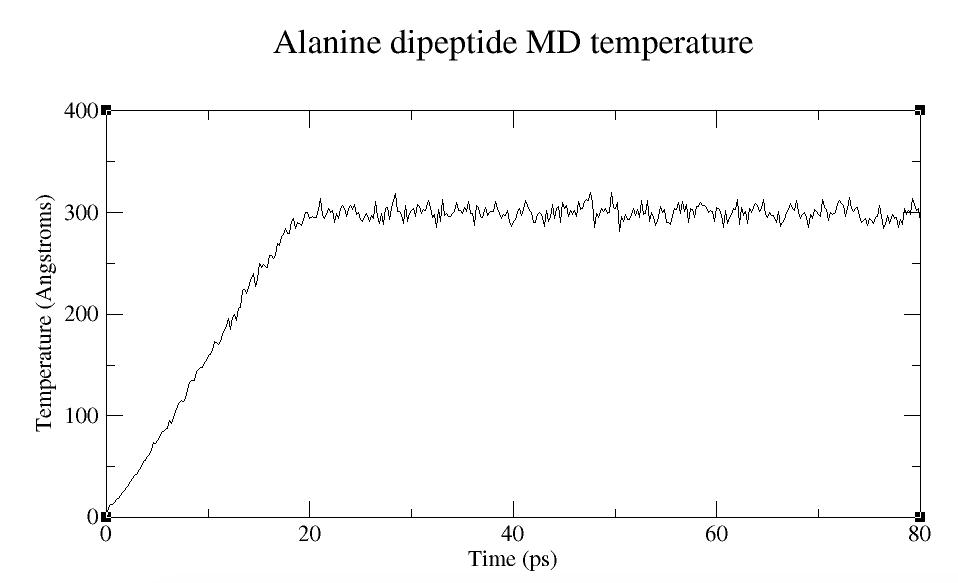
$ xmgrace summary.TEMP  
$ xmgrace summary.DENSITY  
$ xmgrace summary.ETOT summary.EPTOT summary.EKTOT

**Analysis data files**

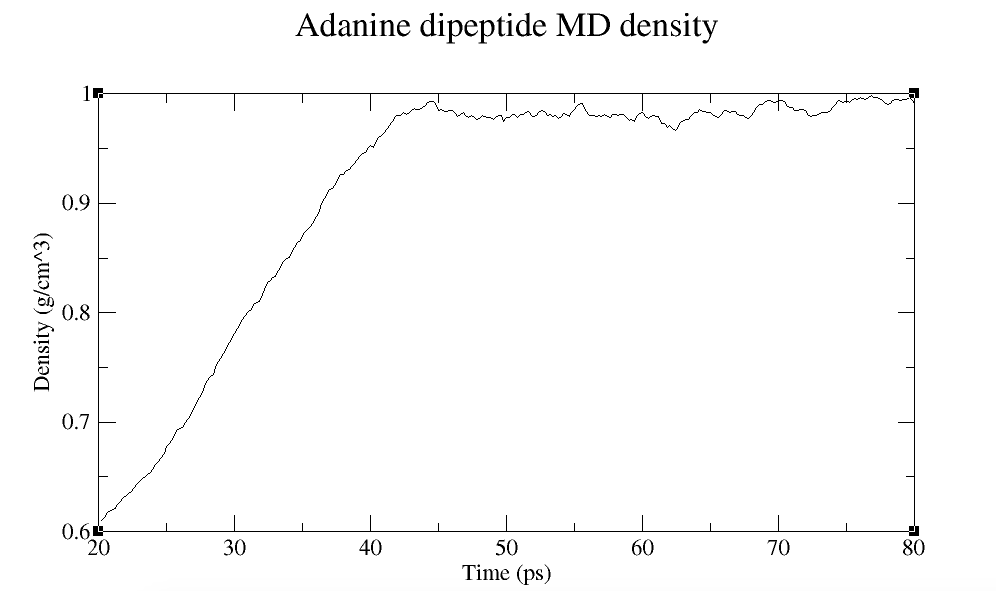
The analysis data files are available here:  
[summary.TEMP](http://ambermd.org/tutorials/basic/tutorial0/include/summary.TEMP)  
[summary.DENSITY](http://ambermd.org/tutorials/basic/tutorial0/include/summary.DENSITY)  
[summary.ETOT](http://ambermd.org/tutorials/basic/tutorial0/include/summary.ETOT)  
[summary.EPTOT](http://ambermd.org/tutorials/basic/tutorial0/include/summary.EPTOT)  
[summary.EKTOT](http://ambermd.org/tutorials/basic/tutorial0/include/summary.EKTOT)

**Production MD system properties**

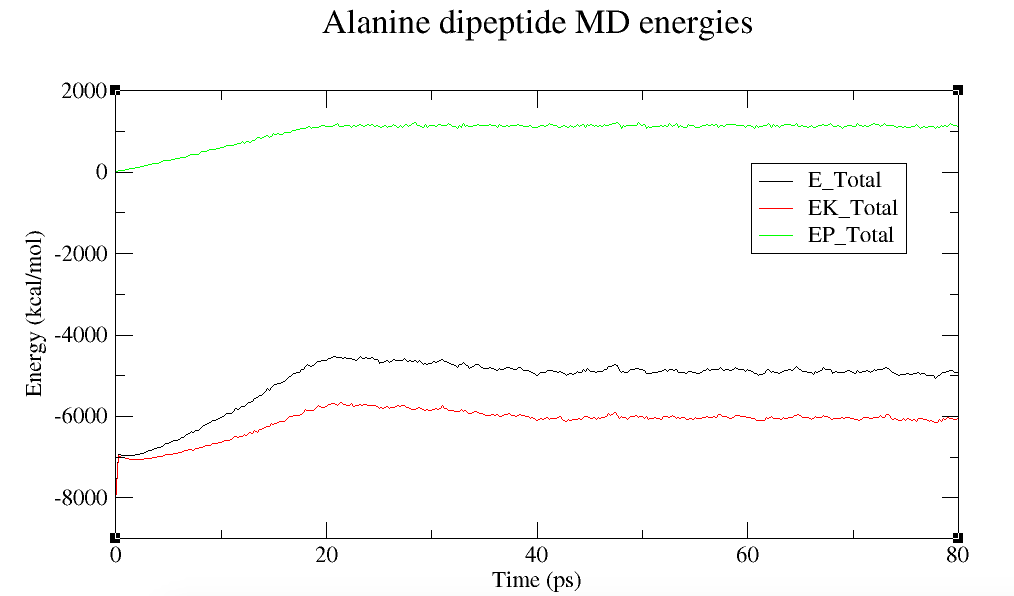
Warning:  
We should run this simulation much longer so that the density has equilibrated and the simulation converges. However, in the interest of time for this tutorial, the production MD simulation time has been set very low so that it possible to analyze the results.   
  
The resulting plots should look similar to these:  
  
**Alanine dipeptide MD temperature**



Here you can see the linear increase in temperature during heating (0-20ns). This is followed by the relatively stable temperature fluctuations about 300 K during the production simulation.  
  
**Alanine dipeptide MD density**



During the 20-80ps, the density equilibrates to approximately 1 g/cm^3. This corresponds to a change in periodic box dimensions as the density of the system converges.  
  
**Alanine dipeptide MD total, potential, and kinetic energy**



This plot shows the total system energy which can be decomposed to the total potential energy and the total kinetic energy.

**Use cpptraj to analyze the RMSD**

The root mean squared deviation (RMSD) value is a measurement of how similar a structure's internal atomic coordinates are relative to some reference molecule coordinates. For this example, we will measure how the internal atomic coordinates change relative to the minimized structure. Specifically, we will analyze the alanine atoms (residue 2).  
  
To do this analysis, we will use **cpptraj**, a fairly comprehensive analysis program for processing MD trajectories. This program runs simple user-written scripts that choose what trajectories to load, what analysis to run, and what processed trajectories or structure to save.  
  
First, we will need to write a simple **cpptraj script** to do this analysis.  
  
37. Use gedit to create a cpptraj script called **rmsd.cpptraj**.

trajin 02\_Heat.mdcrd  
trajin 03\_Prod.mdcrd  
reference 01\_Min.rst  
autoimage  
rms reference mass out 02\_03.rms time 2.0 :2

This is a brief summary of the **cpptraj** script:

|  |  |
| --- | --- |
| trajin 02\_Heat.mdcrd | Load the trajectory 02\_Heat.mdcrd |
| reference 01\_Min.rst | Define the structure 01\_Min.rst as the reference structure |
| center :1-3 mass origin | Center the residues 1-3 by mass to the system origin |
| image origin center | Image the atoms to the origin using the center of mass of the molecule |
| rms reference mass out 02\_03.rms time 2.0 :2 | Calculate the mass weighted RMSD using the reference and output to 02\_03.rms |

**Cpptraj input script file**

The cpptraj input script file is available here:  
[rmsd.cpptraj](http://ambermd.org/tutorials/basic/tutorial0/include/rmsd.cpptraj)

**Run cpptraj**

To actually run cpptraj, we must run it again from the terminal in the directory where the **prmtop**, **mdcrd**, and reference **rst** file is located.  
  
38. Using the terminal, change directory to your Tutorial folder, and run **cpptraj**. A **prmtop** file and **cpptraj script**must be specified.

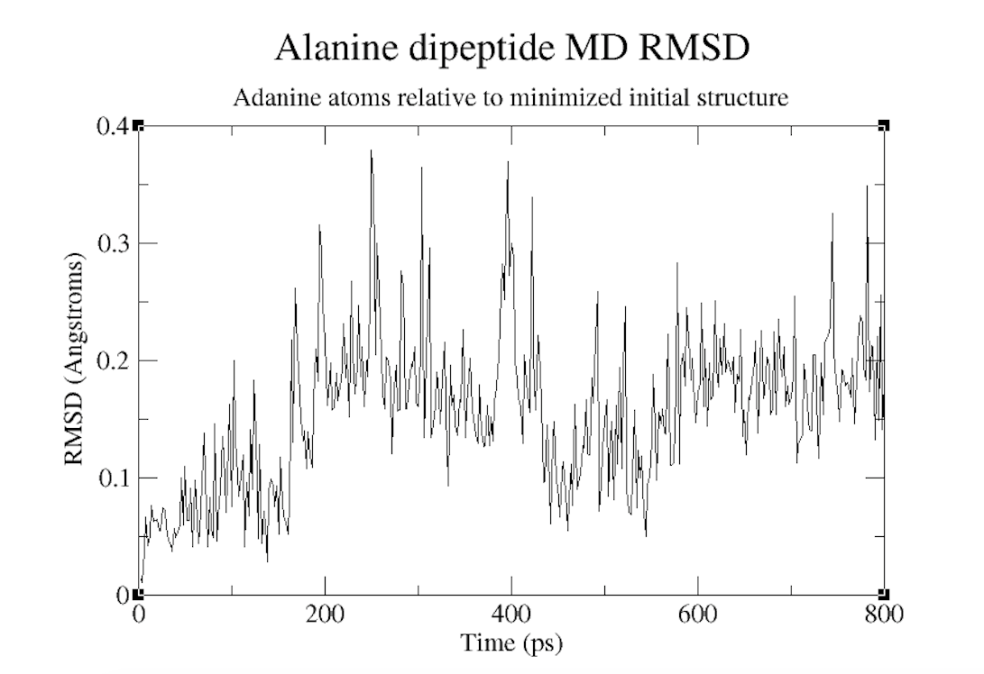
$ cd ~/Tutorial  
$ $AMBERHOME/bin/cpptraj -p prmtop -i rmsd.cpptraj &> cpptraj.log

Now our RMSD data is stored in the file **02\_03.rms**. We can simply plot this file with **xmgrace**.  
  
39. Plot the RMSD with **xmgrace**.

$ xmgrace 02\_03.rms

**Cpptraj output files**

The **cpptraj** output files are available here:  
[02\_03.rms](http://ambermd.org/tutorials/basic/tutorial0/include/02_03.rms)  
[cpptraj.log](http://ambermd.org/tutorials/basic/tutorial0/include/cpptraj.log)  
  
**Alanine dipeptide MD RMSD relative to minimized initial structure**

****

In this example, there is no significant conformational change in the n the Phi/Psi dihedral angles in the alanine dipeptide. This indicates a more stable peptide structure.

**Conclusion**

Congratulations. You've now run your first complete MD simulation and successfully analyzed the results. This is a fairly simple example of the workflow for setting up, running, and analyzing your own MD simulation. If you want to learn more you are encouraged to complete the additional tutorials on the AMBER website.

**Appendix: Files**

All of the files for the tutorial are available here:  
[Alanine\_Dipeptide\_Files.zip](http://ambermd.org/tutorials/basic/tutorial0/include/Alanine_Dipeptide_Tutorial_Files.zip)

Copyright Ross Walker & Ben Madej, 2015