A Recipe for Performing Molecular Dynamics Simulations with Gromacs

Cluster Computing Group, Earlham College
http://cluster.earlham.edu
mailto:ccg@cs.earlham.edu

v2.0 - November, 2007

Overview

This document describes the steps associated with using the Gromacs software package to do molecular dynamics. It also contains a fairly basic description of the science involved. If you need to learn more about the underlying mathematics, physics, chemistry, computer science, and/or biology which support molecular dynamics you should consult one or more of the resources found in Appendix A.

This document assumes you are using one of Earlham College's Cluster Computing Group's (CCG) computational resources to do your work. If not you will need to translate the paths, command names, *etc.* to your environment.

Molecular Modeling

Molecular modeling is the study of molecular structure and function through model building and computational methods. Molecular dynamics is a computational method that solves Newton's equation of motion for all of the atoms in a molecule. At each point in time 6 quantities are known for each atom, position x_p, y_p, z_p and force x_f, y_f, z_f.

Each simulated time-step involves computing the forces on every atom and integrating them to update their positions. The forces are from bonds and electrostatic forces between atoms within a certain cut-off distance. The desired properties are usually obtained as statistical mechanical averages of the atom trajectories over many runs. The averages tend to converge slowly with the length of the simulation or the size of the molecular system.

Molecular dynamics simulations are very computationally intensive, until recently it hasn't been practical to compute more than a couple of nanoseconds for systems with relatively few atoms. Recent developments in cluster and distributed computing algorithms and hardware now make it

possible to (relatively) efficiently and easily harness hundreds or even thousands of processors as part of a single ensemble simulation.

One of the more common uses of molecular modeling/dynamics is to simulate the self assembly of amino acid necklaces (long chain molecules) into proteins, *i.e.* protein folding.

Protein Folding

Proteins are the basis of how the human body builds many of the parts we are made from, *e.g.* enzymes, structural components, and antibodies. Proteins are made out of amino acids, there are 8 essential amino acids which can only be obtained through diet and 14 more which can be manufactured by the body with proper nutrition. DNA contains the recipes for how to arrange which and how many amino acids into a chain, which will then fold into a particular protein. There are many, many possible arrangements, that is 22 amino acids taken in varying numbers and orderings.

Protein folding is a good example of consilience between mathematics, physics, chemistry, computer science, and biology. Sequencing the human genome gave us blueprints for all of the amino acid necklaces, which in-turn fold into proteins which have function within the body. There are connections between the various disciplines at multiple levels.

Why study protein folding? The process is integral to all of biology yet it remains largely a mystery, for example when proteins mis-fold they may be the cause of diseases, *e.g.* Alzheimer's, ALS (Lou Gehrig's), and Parkinson's. Protein structure is also an important component of drug design since potential docking sites must be examined. Protein structure is also a key component of gene therapy research.

Why study protein folding computationally? Folding proteins *In-vitro* is time consuming and expensive compared to folding them *in-silica*. Due to the very short time scales involved *in-vitro* only allows you to see the final protein conformation, not any of the intermediate conformations. In the past it was difficult to marshal enough computational resources to do this but the advent of commodity clusters, commonly known as Beowulf clusters, there are now enough compute cycles within reach of most researchers and educators to fold proteins and other large biomolecules efficiently.

The Software

Groningen Machine for Chemical Simulations (Gromacs, `http://www.Gromacs.org`) is an open source software package originally developed by Groningen University's department of Biophysical Chemistry. Gromacs simulates the forces and movements of atoms in molecular systems over time in a biomolecule such as a protein, *i.e.* molecular dynamics. Gromacs also works well with other large biomolecules such as lipids and even polymers.

The CCG maintains a number of Gromacs installs, you will probably want to use the most recent

version built with the Fastest Fourier Transform in the West (FFTW, `http://www.fftw.org`). If you are using the BobSCEd cluster the binaries for this version of Gromacs are in `/cluster/bobsced/bin` which should be in your default path.

There are a number of tools available for visualizing the results of a molecular dynamics run, either as a movie of the entire run or as an image of the final conformation. The CCG uses PyMol (`http://pymol.sourceforge.net`, or `http://delsci.com/macpymol` for a pre-built version for OS X) for most of our work, it's capable of both movies and still images. A Google search using "visualize molecular dynamics" will yield many other visualization software choices.

The Process

The molecular system used in this example, 2LZM, is a lysozyme from bacteriophage T4. The commands described below use particular force fields, etc. appropriate for that system. Depending on the molecular system you are working with other choices may be more appropriate.

All of the Gromacs programs have built-in descriptions of their command line arguments, to see these type `<program-name> -h` at the command line. This is the best way to start learning about the many choices Gromacs offers. The instructions that follow are patterned after Lindahl[x].

Download the Molecular System

The Research Collaboratory for Structural Bioinformatics (RCSB) provides the Protein Data Bank at `http://www.rcsb.org`. Typing in 2LZM as the PDB ID at that site returns a reference to a particular molecule, Lysozyme. Selecting "Display File" and then downloading an uncompressed PDB yields a file on the local machine which gives coordinates of every atom in the molecule. You will need to upload that file to the CCG file system using `scp` or `ftp` from your local machine, the host name to use is `cluster.earlham.edu` along with your CCG username and password.

Convert to Gromacs File Formats

`pdb2gmx` will convert the description of the system from the PDB format to a full topology file and a coordinate file. The OPLS-AA/L force field is used.

```
$ pdb2gmx -f 2lzm.pdb -o 2lzm.gro -p 2lzm.top -i 2lzm.itp -ff oplsaa
```

This command creates three files: `2lzm.gro`, the coordinates of the atoms in the molecule, `2lzm.top`, the topology, and `2lzm.itp`, the position restraint data. At each time step the simulation will compute the forces between each atom in the system.

Place the Molecule in a Box full of Water

For this simulation to take a reasonable amount of time we restrict the size of the system by placing the molecule in a box. Only forces within this box are calculated. To restrict a cubic box to a size of 0.5nm to the box's edge from any atom in the molecule we run the command:

```
$ editconf -f 2lzm.gro -d 0.5 -bt cubic -o 2lzm-box.gro
```

At this point the box is empty save the molecule. In order to create a more realistic simulation we fill the empty space in the box with the SPC 216 water model using `genbox` and output the resulting system in PDB format:

```
$ genbox -cp 2lzm-box.gro -cs spc216.gro -p 2lzm.top -o 2lzm-solvated.pdb
```

Minimize the Energy in the System

Next we use energy minimization to remove overlapping atoms. This requires preparing the system for a short run with `grompp` and then running it with `mdrun`. First copy the configuration file `/cluster/bobsced/share/gromacs/2lzm-em.mdp` to your working directory. This file contains the parameters required for the different steps in preparing and running the energy minimization simulation. The `-np 1` parameter to `grompp` and `mdrun` specifies that a single process will be used.

Some of the command lines given below do not fit on a single line, `$` denotes the beginning of a command line, if there are one or more arguments without a leading `$` they are a part of the line above them.

```
$ cp /cluster/bobsced/share/gromacs/2lzm-em.mdp .
$ grompp -f 2lzm-em.mdp -p 2lzm.top -c 2lzm-solvated.pdb -o 2lzm-em.tpr
$ mdrun -np 1 -v -s 2lzm-em.tpr -o 2lzm-em.trr -c 2lzm-em.gro -e 2lzm-em.edr
  -g 2lzm-em.log -x 2lzm-em.xtc
```

The next step further prepares the system by holding the target molecule stable but allowing the water to settle around it, this process is called position restraining. To do this you will need to copy the MD parameter file `/cluster/bobsced/share/gromacs/2lzm-pr.mdp` to your working directory. Again we'll use `mdrun` to run the simulation once the system is prepared.

```
$ cp /cluster/bobsced/share/gromacs/2lzm-pr.mdp .
$ grompp -np 1 -f 2lzm-pr.mdp -p 2lzm.top -c 2lzm-em.gro -o 2lzm-em-pr.tpr
$ mdrun -np 1 -v -s 2lzm-em-pr.tpr -o 2lzm-em-pr.trr -c 2lzm-em-pr.gro
  -e 2lzm-em-pr.edr -g 2lzm-em-pr.log -x 2lzm-em-pr.xtc
```

Run the Molecular Dynamics Simulation

Finally we are ready to run the full molecular dynamics simulation. This involves using a different set of configuration options for `mdrun` which are found in `/cluster/bobsced/share/gromacs/2lzm-md.mdp`. Again we'll prepare the system for the simulation with `grompp`.

```
$ cp /cluster/bobsced/share/gromacs/2lzm-md.mdp .
$ grompp -np 1 -f 2lzm-md.mdp -p 2lzm.top -c 2lzm-em-pr.gro -o 2lzm-em-pr-md.tpr
$ mdrun -np 1 -v -s 2lzm-em-pr-md.tpr -o 2lzm-em-pr-md.trr -c 2lzm-em-pr-md.gro
  -e 2lzm-em-pr-md.edr -g 2lzm-em-pr-md.log -x 2lzm-em-pr-md.xtc
```

Visualize the Results of the Simulation

In order to visualize the system that has just been simulated we use Pymol, which can view both the final conformation and animate the entire simulation run. To use the output of Gromacs with Pymol we must first remove the water from the system and extract the protein coordinates, etc. in a form Pymol can read, `trjconv` does this for us.

```
$ g_filter -s 2lzm-em-pr-md.tpr -f 2lzm-em-pr-md.xtc -ol lowpass.xtc -nf 10 -all
$ trjconv -s 2lzm-em-pr-md.tpr -f lowpass.xtc -o 2lzm-final.pdb
  Select Group 1 to convert just the Lysozyme molecule
```

The easiest way to use Pymol is to install the appropriate package on your client machine (running OS X, X/Un*x, or Windows) and then download and view the resulting PDB file, `2lzm-final.pdb` in this example, on the client.

Where to Go from Here

Gromacs has many other programs which *e.g.* measure the energy in the system (`g_energy`) and calculate the root mean square displacement of the system (`g_rms`). The Gromacs manuals and Lindahl[x] are good sources of information about these and other Gromacs utility programs.

Appendix A - Resources

BibTeX this.

1) Gromacs' manuals and tutorials `http://www.gromacs.org/content/view/13/27`.

2) Kristina's DNA overview in the SC Education wiki.

3) CCG's Gromacs Introduction curriculum module.

4) CCG's Gromacs peptide curriculum module.

5) CCG's Gromacs gmxbench curriculum module.

6) Lindahl, Erik. Parallel Molecular Dynamics: Gromacs. Cluster Monkey, August, 2006. `http://www.clustermonkey.net//content/view/141/33`

7) Berman's PDB article.

8) DeLano, W.L. The PyMOL Molecular Graphics System (2007) DeLano Scientific LLC, Palo Alto, CA, USA. `http://www.pymol.org`

Appendix B - To Do

1) Document running Gromacs in parallel with LAM-MPI and MPICH. Gromacs now supports domain decomposition controls and threading.

2) Document the most the efficient way to leverage:

- Multi-core machines
- SMP machines with shared memory
- Constellations (multi-core clusters, SMP clusters, MC+SMP (*ala* BobSCEd))

This depends in part on the size of the system in question (the number of mass points) and the long range interaction types, as well as the architecture of the underlying compute resource.

3) Document using Gromacs on TeraGrid compute resources (`http://teragrid.org`).

4) Build a proper bibliography (see Charlie's dissertation).