

## 1 Introduction

GROMACS (**GRO**ningen **MA**chine for **C**hemistry **S**imulation) is a molecular dynamics (MD) simulation package originally developed at the University of Groningen. The highly optimized code makes GROMACS the fastest program for molecular simulations to date.

The source code for GROMACS is freely available at <http://www.gromacs.org>; there you will also find complete documentation as well as a trouble-shooting guide. A wiki page is also available at <http://wiki.gromacs.org>.

GROMACS is invoked via several command-line options; there is no user-interface, per se. GROMACS is a program for the Unix environment. Please visit the GROMACS website for a complete list of functions that you can invoke from the command line. Detailed information about all of the routines that are used in this tutorial can be found online. A simple Google<sup>TM</sup> search will suffice. The following GROMACS routines are invoked in this tutorial: `pdb2gmx`, `editconf`, `genbox`, `grompp`, `mdrun` and `ngmx`.

The contents of this tutorial assume that you are working on the beowulf computer cluster at `csc.smith.edu`. Also, the `grompp`, `mdrun` and `pdb2gmx` commands are not explained fully in this tutorial, because it is assumed that you've completed the first tutorial, where they were explained in more depth. Please reference that tutorial if you have any questions. GROMACS has been installed on the beowulf cluster, and the library of GROMACS functions are all available; you do not have to install GROMACS. Throughout this tutorial, text that is intended to be typed at the console is written within a box, such as:

`command intended for command prompt`

The following files are being provided, and it is assumed that they are in a directory `md_tutorial`. If they are located in a different directory, you'll need to adjust the tutorial accordingly whenever a reference is made to `md_tutorial`:

- `emTutorial2.mdp` - parameter file for the energy minimization MD step
- `prTutorial2.mdp` - parameter file for the position restrained MD step
- `mdTutorial2.mdp` - parameter file for the main MD step
- `1ako.pdb` - a PDB file
- `1arl.pdb` - a PDB file
- `1hhp.pdb` - a PDB file

## 2 Summary of Tutorial

This tutorial is composed of the following sections:

1. Choosing a protein
2. Determining the location of the catalytic site
3. GROMACS preliminaries - is it installed?
4. Creating a topology file for pre-processing and outputting position restrained information
5. Creating and solvating a simulation box
6. Energy minimization - the solvent
7. Position restrained MD - filling in the spaces
8. Creating parameter files for position restrained MD run
9. Performing the MD simulation and constraining most of the molecules EXCEPT those that are part of the active site
10. Viewing the MD Simulation Results

## 3 MD Tutorial

### 3.1 Choosing a protein; using PyMOL

You first must select a protein (or protein fragment) on which you want to perform MD. PyMOL (<http://pymol.org/>) is a freely available molecular viewing program that has been installed on the beowulf cluster; use it to view several proteins, one of which you'll select for your MD simulation.

The protein data bank (PDB, [www.pdb.org](http://www.pdb.org)) is a repository of X-ray crystallographic data of proteins. Figure 3.1 lists the proteins that have been retrieved from the PDB and which have been saved to the beowulf cluster.

When you have logged into your Linux account, navigate to `md_tutorial` directory; there you will see the saved PDB files as well as three files `em.mdp`, `md.mdp`, and `mdout.mdp` that are used in this tutorial. Be sure to log into your Unix account, and not your Windows account. Next, invoke PyMOL from the command prompt:

```
pymol
```

When you open and view the protein, use the mouse to rotate the protein, to zoom in and out, to perform measurements, etc. PyMOL offers a wide range of features, similar to many other protein viewing programs; explore these features at your leisure. While you are using PyMOL, take some time to ask some questions about the protein that you are viewing. For example:

- What are the prominent features of the protein?
- Where do you think the protein will exhibit motion (if any)?

PDB ID	Protein Size	Description
1ako	268 Residues	Exonucleases are enzymes that cleave nucleotides one at a time from an end of a polynucleotide chain. Exonucleases are involved in many cell processes, including the clipping of the RNA primer region during DNA synthesis. 1ako is the exonuclease from <i>escherichia coli</i> , a bacteria.
1arl	307 Residues	1arl is a carboxypeptidase, which hydrolyzes peptide bonds of C-terminal residues with aromatic or aliphatic side chains.
1hhp	99 residues	HIV1 Protease is a protease that hydrolyzes HIV viral polyproteins into functional protein products that are essential for viral assembly and subsequent activity of HIV.

Figure 1: The 3 PDB files have been selected specifically for this lab because of their appropriate length, which is conducive to an MD simulation that can be performed in a short amount of time.

- Are certain parts of the protein “clumpy”, while others less so?

This tutorial will perform Molecular Dynamics on 1HHP. If you choose to perform an MD simulation on a protein other than 1HHP, you’ll need to adjust the tutorial accordingly.

### 3.2 Determining the location of the catalytic site

The catalytic site of a protein is that location where a protein performs catalysis. Often-times, the functional part of the protein is called the “active site”, because that is where all of the interesting “activity” takes place; that’s not to say that the other parts of the protein are not moving, but rather that it is at the active site that the “important” events occur. And, because scientists are most interested in the active site of a protein, we are interested in simulating the motion of the atoms at – or at least on all atoms within a certain distance – of the active site.

The website <http://www.ebi.ac.uk/thornton-srv/databases/CSA/> provides active site information for many proteins that are in the PDB. Navigate to the website and enter “1HHP” into the field marked “PDBCode” and press the “Search” button. One part of the results should be the following:

<input checked="" type="checkbox"/> Catalytic Site (Get help with this section)				
Found by: PsiBLAST alignment on <a href="#">1a30</a>		Residue	Chain Number	UniProt number
		ASP	A 25	25
		THR	A 26	26
		Functional part		
		Unknown		

<input checked="" type="checkbox"/> Catalytic Site (Get help with this section)				
Found by: PsiBLAST alignment on <a href="#">1hiv</a>		Residue	Chain Number	UniProt number
		ASP	A 25	25
		Functional part		
		Unknown		

So, the catalytic explorer found, depending on which algorithm was used, that 1hhp has an active site at residue number 25 and 26. So, what we’ll do in in this tutorial is perform Molecular Dynamics on the regions around those residues. As is often done in the laboratory setting, molecular dynamics is performed only on that region of a protein that is known to contain the active site. That way,

all of the “non-important” parts of the protein need not be simulated, which can speed up the simulation significantly.

### 3.3 GROMACS preliminaries - is it installed?

Before you can proceed, you need to make sure that all of the routines in GROMACS are available and that they are in your path. While logged into your lab Unix account, invoke the following:

```
which mdrun
```

If necessary, add the GROMACS functions to your path:

```
set path = ($path /usr/local/gromacs/bin)
```

Please refer to the first tutorial which gives more detailed instructions on how to make sure that the GROMACS functions are in your path.

Once you’ve confirmed that the GROMACS routines are available for you to use, you’ll want to copy the PDB file that you want to simulate in GROMACS to a common name that coincides with the name of the protein that is used in the tutorial. The following commands create a copy of 1HHP.pdb and move all of the necessary files into a mdrunRestrained directory:

```
cp 1HHP.pdb protein.pdb
```

```
mkdir mdrunRestrained
```

```
cp * mdrunRestrained/.
```

```
cd mdrunRestrained
```

### 3.4 Creating a topology file for pre-processing

Before you run the MD simulation, we have to tell **mdrun** to **restrain** the portion of those atoms that are not part of the active site. To run an MD simulation, you need to prepare several files. These files are what you use to indicate to GROMACS exactly how you want to perform the simulation (for how long), and how you want to set up the simulation environment (should the protein be in water? in alcohol?, etc).

First you need to create a molecular topology file that is used by GROMACS. Again, refer to the first tutorial for more detailed information on this step:

```
pdb2gmx -f protein.pdb -o protein.gro -ff G43a1 -p protein.top -i posre.itp
```

### 3.5 Creating and solvating a simulation box

Biological structures such as proteins are not just “floating” around in space, and so you must solvate your peptide in a box of water; this will make the simulation a bit more “realistic”. Invoke the editconf and genbox commands to create a simulation box and to solvate the box with

water molecules. Again, please refer to the first tutorial for a more detailed explanation on this step:

```
editconf -f protein.gro -o protein.gro -d 0.85
```

```
genbox -cp protein.gro -cs -o protein_b4em.gro -p protein.top
```

### 3.6 Energy minimization - the solvent

So, at this moment you’ve “dumped” water into your simulation box, which surely has caused some close contacts that are energetically unfavorable. The standard procedure to remove these close contacts is Energy Minimization (EM). Energy minimization perturbs the coordinates of individual atoms until the very high repulsive energies are eliminated (or at least reduced as much as possible). Before you can start the EM step, you need to pre-process all the input files with the GROMACS preprocessor function **grompp**, and then you can invoke the **mdrun**:

```
grompp -f emTutorial2 -c protein_b4em -p protein -o protein_em
```

```
mdrun -nice 4 -s protein_em -o protein_em -c protein_b4pr -v
```

### 3.7 Position restrained MD - filling in the spaces

Once all close contacts are removed from the system, you’ll want to do molecular dynamics of the water molecules, and keep the peptide fixed. This is called position restrained (PR) MD, and the purpose is to let all water molecules equilibrate around the peptide in order to fill holes, crevices, etc. which were not filled by the **genbox** program. You first need to generate the topology file using the **grompp** program, and then you can invoke the **mdrun** routine:

```
grompp -f prTutorial2 -c protein_b4pr -r protein_b4pr -p protein -o protein_pr
```

```
mdrun -nice 4 -s protein_pr -o protein_pr -c protein_b4md -v
```

### 3.8 Creating parameter files for position restrained MD run

Prior to performing the lengthy MD simulation, we want to create the parameter file as well as the position restrained file that indicates to **mdrun** which molecules should and which molecules should not be restrained.

First, we are going to modify the original Position Restrained file (**posre.itp**) to restrain motion on those residues that are NOT part of the active site. According to the Catalysis Site Atlas, protein 1HHP has an active side at residues 25 and 26. To be safe, let us allow motion on all residues OTHER than 24-27.

Open and view the **protein.top** file. The topology file is the GROMACS format of the original data that was contained in the PDB, as well as additional data for bonds, Salt Bridges, etc. What you want to do is locate those atoms that are part of residues 24-27. Scroll down to the section that is labeled as [ **atoms** ] and locate those atoms that are part of residues 24-27. A snippet of the data is provided here:

229	CH3	23	LEU	CD2	98	0	15.035	; qtot 3
230	C	23	LEU	C	99	0.38	12.011	; qtot 3.38
231	O	23	LEU	O	99	-0.38	15.9994	; qtot 3
232	N	24	LEU	N	100	-0.28	14.0067	; qtot 2.72
233	H	24	LEU	H	100	0.28	1.008	; qtot 3
234	CH1	24	LEU	CA	101	0	13.019	; qtot 3
235	CH2	24	LEU	CB	101	0	14.027	; qtot 3
236	CH1	24	LEU	CG	102	0	13.019	; qtot 3

Notice that residue number 24 begins with atom 232. Likewise, if you scroll down further, you'll notice that the last atom of residue 27 is atom 263. So, because we want to restrain the motion of residues 24 through 27, we must restrict the motion of atoms 232 through 263.

Open up the file `posre.itp`, which is the original position restrained file that was generated by the original `pdb2gmx` program. `posre.itp` lists all of the heavy atoms (i.e. backbone and side chain atoms) in the original PDB. If you are using emacs, a common editor, you would input:

```
emacs posre.itp
```

The format of the `posre.itp` is as following:

```
; atom  type      fx      fy      fz
      1      1    1000    1000    1000
      4      1    1000    1000    1000
```

where the `fx`, `fy`, and `fz` terms designate the axis that is being constrained for each indicated atom. Therefore, if you wanted to constrain the motion of atom 1, for example, to only motion in the *y* coordinate, you would change the line correspond to atom 1 to the following:

```
; atom  type      fx      fy      fz
      1      1    1000        0    1000
      4      1    1000    1000    1000
```

which indicates that atom 1 can move in the *y* directions because there is no “constraint” placed on that axis. Thus, to REMOVE the constraints on atoms 232-263, you'll need to change all of the `fx`, `fy` and `fz` values for those atoms to 0, or simply erase those lines in the file, because any atom that is NOT contained in the position restrained file has full motion. So, we are going to delete the rows in the position restrained file that correspond to those atoms that belong to the active site that we want to simulate, so you you'd want the relevant part of `posre.itp` as the following:

```
230      1  1000  1000  1000
231      1  1000  1000  1000
264      1  1000  1000  1000
266      1  1000  1000  1000
```

Once you've done that, you are ready to invoke the gromacs preprocessor:

```
grompp -f mdTutorial2 -c protein_b4md -p protein -o protein.md
```

### 3.9 Performing the MD simulation and constraining most of the molecules EXCEPT those that are part of the active site

Once you've executed the `grompp` program in the previous step, you are ready to invoke the actual MD simulation, which, according to the `mdTutorial2.mdp` file, is set to simulate 20ps of motion:

```
mdrun -nice 4 -s protein.md -o protein.md -c protein_after.md -v
```

### 3.10 Viewing the MD Simulation Results

To view the MD results, invoke the `ngmx` program:

```
ngmx -f protein.md -s protein.md
```