Membrane protein tutorial with GROMACS

Yiannis Galdadas, George Patargias, and Zoe Cournia
Biomedical Research Foundation, Academy of Athens

1) Introduction

This tutorial describes a series of steps to set up and run an MD simulation of a membrane protein embedded in a solvated lipid bilayer using the GROMACS program and its associated tools. The tutorial makes uses of GROMACS v.4.5.5 as the necessary for the embedding process "g_embed" tool is not (yet) fully functional in GROMACS v.5.0 or higher. Please feel free to contact us for any question regarding the tutorial, for improvements/corrections on the tutorial or if any bug is found.

Before installing GROMACS, the installation of FFTW libraries is needed

- Download FFTW 3.3 source from here
- Unzip and untar the fftw-3.3.tar.gz file and cd to the created fftw-3.3 dir
- To compile a single precision of the libraries, enter:

```
./configure  --enable-threads –enable-float --enable-shared
make
make install
```

Note: If your machine is i686 or x86-64, add --enable-sse in the ./configure command line. By default, the header files will be installed in /usr/local/include and the library files in /usr/local/lib. If you do not like that (since you must have root permissions to write there), you can place them e.g. in your own home directory by using the option –prefix in the ./configure command line. As an example, if you set --prefix=/home/george/fftw, the headers will go in /home/george/fftw/include and the library files in /home/george/fftw/lib.

To repeat the compilation/installation with a changed ./configure command line, enter first

```
make distclean
```

To install GROMACS on a Linux machine:

- Download the source for 4.5.5 version from here
- Unzip and untar the GROMACS-4.5.5.tar.gz file and cd to the created GROMACS-4.5.5 dir
- Define the location of the FFTW headers/libraries (if they are not at the library path).
- export CPPFLAGS="-I/home/george/fftw/include"
- export LDFLAGS="-L/home/george/fftw/lib"
- Run the configure script
- ./configure –-enable-threads –prefix=/home/george/GROMACS
- This will install GROMACS in /home/george/GROMACS/lib and /home/george/GROMACS/bin
- Compile and install by entering
- make

- make install

The above instructions are for installing and running GROMACS on a multi-core desktop ("single-node"). If you want to repeat compilation/installation with a different ./configure command line, first, clean your disk using the following command:

make distclean

The use of the GROMACS programs/tools (pdb2gmx, editconf, genbox etc), all located in $HOME/GROMACS/bin, to set up the simulation system is shown below. A brief description of what each command does including input/output files and flags can be obtained by typing the command with the -h flag. E.g. if we want to know what the command editconf does we give:

editconf -h

After installing GROMACS, unzip and untar the contents of the file memprot_tutorial.tar.gz:

unzip memprot_tutorial.zip

and cd to the created tutorial_files/ directory.

2) Generate topologies

Note1 before we begin: If you try to copy/paste the commands given below from the Word file to your terminal, in most of the cases, you may end up with an error warning you that the tool you want to use does not exist, or that the files that you want to pass as inputs do not exist etc. So we encourage you to type the commands yourself.
Note2 before we begin: In Gromacs versions before 5.0.* the sequence of the commands does matter, i.e. inputs come before outputs, so pay attention to following the order of the flags in the following commands.

So let's begin!

Download the Aquaporin 1 (AQP1) structure from PDB (pdb code 1H6I) -> 1h6i.pdb (already present in the memprot_tut_files/ directory). A pdb file contains in a standardized format the atom names/numbers, the amino acid names/numbers and the x, y, z coordinates of any protein structure.

Make a copy of the downloaded 1FX8.pdb file,

cp 1h6i.pdb aqp1_prot.pdb

In aqp1_prot.pdb, remove all lines except the ATOM records.

Run the GROMACS pdb2gmx command to generate the protein topology and add the missing hydrogens to the protein, This program also takes care of the protonation state of titratable residues (ASP, GLU, etc). By default, these will be set to the charged state (normal for pH=7) but it is possible to do this interactively for each one of these residues in the protein. E.g. using the -arg flag, pdb2gmx will ask the user about the preferred state of every single ARG residue in the protein. Similarly, with -asp, -glu, -lys flags.

pdb2gmx -f aqp1_prot.pdb

Select option 14 (GROMOS96 53a6 force field)

Select none for water model (we will do this later)

Note the protein charge ("Total charge 1.000 e")

From pdb2gmx we obtain 3 files:

conf.gro - contains the protein structure with added hydrogens in GROMOS format (instead of pdb)
topol.top - contains the protein topology and #include directives
posre.itp - a file with the position restraints

Make a file with the protein topology only

cp topol.top protein.itp

In protein.itp, remove all lines in the beginning until [moleculetype] and at the end all lined under the following #endif

#ifdef POSRES
#include "posre.itp"
#endif

We have downloaded the following files from Peter Tieleman's website.

dppc128.pdb - the structure of an equilibrated 128-lipid DPPC bilayer
dppc.itp - the moleculetype definition for DPPC
lipid.itp - Berger lipid parameters (contains all the atom types, non-bonded parameters, and bonded parameters for a large class of lipids)

In the memprot_tut_files/ directory we have made a sub-directory called "gromos53a6_lipid.ff/" and copied the files from the GROMACS distribution /usr/local/GROMACS/share/GROMACS/top/gromos53a6.ff/ into it.

In the memprot_tut_files/ directory, we have modified the ffnonbonded.itp and ffbonded.itp files so as to include the Berger lipid parameters contained in the lipid.itp file following the same procedure as the one described in the following tutorial (http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/membrane_protein/02_topology.html).

Make a file called system.top for the system's topology that looks like this:

#include "gromos53a6_lipid.ff/forcefield.itp"
#include "protein.itp"
#include "dppc.itp"

```
[ system ]
; Name
AQP1

[ molecules ]
; Compound        #mols
Protein_chain_A     1
```

Note: The above #include statement has a similar function with #include statement in C. It tells the GROMACS preprocessor to copy the contents of the files forcefield.itp (forcefield parameters), protein.itp (protein topology), dppc.itp (lipid topology) in the place where #include statement is located. "Protein_chain_A" declares the existence of one AQP1 molecule in the system. "Protein_chain_A" is the name of the protein as is declared in [moleculetype] section of the protein.itp file. Note that you can change this name anytime if convenient, but make sure that the protein's name in the .itp and .top files is in accordance.

Convert the dppc128.pdb to dppc128. gro with editconf tool,

```
editconf -f dppc128.pdb -o dppc128.gro
```

Make a topology file named topol_dppc.top which describes the 128 DPPC system and looks like this:

```
#include "gromos53a6_lipid.ff/forcefield.itp"
#include "dppc.itp"
#include "gromos53a6_lipid.ff/spc.itp"

 [ system ]
; Name
128-Lipid DPPC Bilayer

[ molecules ]
; Compound #mols
DPP  128
SOL 3655
```

The #include "gromos53a6_lipid.ff/spc.itp" tells the GROMACS preprocessor to copy the parameters described in the spc.itp and indicates that the SPC three-point water model will be used in the simulation for the treatment of water molecules.

3) Generate a larger bilayer

Run grompp (the GROMACS pre-processor) to get a run input .tpr file. This is a portable binary file that contains all the information that the main MD engine of GROMACS (mdrun) needs to perform the simulations. It contains the starting structure of the simulation (coordinates from the input .gro/.pdb file and velocities generated through a random assignment from a Maxwell-Boltzmann distribution) the molecular topology (in the .top file) and all the simulation parameters (in the .mdp file). Information about all GROMACS input/output files

is here. GROMACS tools need the .tpr file as input due to the important information about the system's coordinates and parameters that it contains. At this step, the .tpr file that we will generate, we won't use it to perform any simulation but to use the GROMACS tools so as to create a larger lipid bilayer.

grompp -c dppc128.gro -p topol_dppc.top -f min.mdp -o min.tpr -maxwarn 1

You can see the contents of the .tpr file in text format using the command gmxdump

gmxdump –s min.tpr > min.tpr.txt

Before replicating the lipid bilayer, remove periodicity from dppc128.gro running the trjconv command and using the min.tpr generated in the previous step.

trjconv -s min.tpr -f dppc128.gro -o dppc128_whole.gro -pbc mol -ur compact
Select group for output: Group 0 (System)

Increase slightly the box dimensions of dppc128_whole.gro,

editconf -f dppc128_whole.gro -o dppc128_whole_boxnew.gro -box 6.5 6.5 6.6

Note: Box dimensions can be visualized in rasmol with the command "set unitcell on".

Replicate the 128 DPPC bilayer to obtain a bigger bilayer of 512 DPPC lipid bilayer that is suitable for embedding a membrane protein. You can do so via VMD using the replicate.tcl script that is provided in the tutorial files. Specifically, load the dppc128_whole_boxnew.gro in VMD and at the VMD prompt in the text console window type,

source replicate.tcl

Save the coordinates of the replicated DPPC bilayer in a .pdb format named *replicate.pdb.*

Update the topol_dppc.top file so as to describe the topology of the replicated DPPC bilayer. The topol_dppc .top should look like this:

```
#include "gromos53a6_lipid.ff/forcefield.itp"
#include "dppc.itp"
#include "gromos53a6_lipid.ff/spc.itp"

[ system ]
; Name
DPPC + water

[ molecules ]
; Compound    #mols
DPP             512
SOL             14620
```

At this step, it is a good idea to reorganize your coordinate file (replicate.pdb) so as all DPPC molecules to be in a row followed by the water molecules. We will do this by generating an index file that will allow us to isolate each component of the DPPC/water system separately.

make_ndx –f replicate.pdb –o index_reorganize.ndx
press q when prompt to

First, we will isolate the DPPC molecules,

editconf –f replicate.pdb –o replicate_dppc.gro –n index_reorganize.ndx
select DPP group when prompt to

Then, we will isolate the water molecules,

editconf –f replicate.pdb –o replicate_water.gro –n index_reorganize.ndx
select SOL group when prompt to

We will now concatenate the two files so as to rebuild the DPPC/water system,

cat replicate_dppc.gro replicate_water.gro > replicate_new.gro

Open replicate_new.gro file with a text editor and remove the three lines between the last lipid atom entry and the first water entry (i.e. the line that contains the box vectors "13.0000 … 43860" of the lipid box, the line with the Gromacs message and the line with the number of water molecules "43860"). Do not forget to change the second line of the replicate_new.gro file from 25600 to 69460 so as to correspond to the sum of the lipid atoms (25600) and water atoms (43860).

Note the line #include "gromos53a6_lipid.ff/spc.itp" which tells GROMACS to take into account the parameters of the SPC water model used in the simulations as described before.

Generate the .tpr file for a short energy minimization of the replicated DPPC bilayer. The parameters for the energy minimization are contained in the min.mdp file,

grompp -c replicate_new.gro -p topol_dppc.top -f min.mdp -o min.tpr

Run the energy minimization,

mdrun -s min.tpr -deffnm em_membrane -v

Use the minimized system coordinates (em_membrane.gro) to run a 100 ps MD simulation. The simulation parameters are contained in the equil_npt.mdp file. Information about the meaning of all these parameters can be found here.

grompp -c em_membrane.gro -p lipid.top -f equil_npt.mdp -o equil_npt.tpr

Run the simulation,

mdrun -s equil_npt.tpr -deffnm equil_membrane_out

You can monitor the progress of the simulation, which is written in the .log file via typing in a new terminal session,

tail –f equil_membrane_out.log

4) Insert the protein into the larger bilayer

Convert the output system coordinates to pdb format.

editconf -f equil_membrane_out.gro -o equil_membrane_out.pdb

Place protein pdb from pdb2gmx in the center of the box of equil_membrane_out.gro, first by running editoncf,

editconf -f conf.gro -o conf_box.pdb -box 12.72316 12.72316 6.52687

and then by concatenating the two pdb files,

cat conf_box.pdb equil_membrane_out.pdb > prot+dppc_tmp.pdb

Note that the new box of the protein after the first step (editconf step) are the same with those of the DPPC/water system as they are defined in the last line of equil_membrane_out.gro.

Remove all the lines of the prot+dppc_tmp.pdb file between the last protein atom entry (starting with “ATOM 2102 O2 PRO 233 83.952 56.902 12.891 1.00 0.00”) and the first lipid entry (staring with “ATOM 1 C1 DPP 1 113.960 15.450 8.470 1.00 0.00”) . Do not leave any blank line between “ATOM 2102 O2 PRO 233 83.952 56.902 12.891 1.00 0.00” and “ATOM 1 C1 DPP 1 113.960 15.450 8.470 1.00 0.00”.

Renumber atom numbers in prot+dppc_tmp.pdb with editconf tool,

editconf -f prot+dppc_tmp.pdb -o prot+dppc.pdb

editconf -f prot+dppc.pdb -o prot+dppc.gro

Update the system.top so as to correspond to the new composition of your system, i.e. the protein placed in the middle of the 512 DPPC lipid bilayer. Namely, include the following lines,

#include "gromos53a6_lipid.ff/forcefield.itp"
#include “protein.itp”
#include "dppc.itp"
#include "gromos53a6_lipid.ff/spc.itp"

[system]
; Name
AQP1 + **DPPC + water**

[molecules]
; Compound #mols
Protein_chain_A 1
DPP 512
SOL 14620

Place the system in a bigger box

editconf -f prot+dppc.gro -o prot+dppc.gro -box 12.72316 12.72316 12

Add water molecules to the void regions of the system,

genbox -cp prot+dppc.gro –cs –o prot+dppc+sol.gro –p system.top

Write down the number of added waters. At the end of genbox output: "Number of SOL molecules: 44278"

Edit system.top correcting the number waters (44278).

Generate a .tpr file needed by the GROMACS g_membed program that inserts the protein into the membrane. For this, we use the provided membed.mdp file.

grompp -c prot+dppc+sol.gro -p system.top -f membed.mdp -o membed.tpr

Run the g_membed program (according to the Gromacs version that you use) and insert the protein into the membrane,

g_membed -f membed.tpr -deffnm membed_out (GROMACS v.4.5.5)
g_membed -f membed.tpr -p system.top -o membed_out.trr -x membed_out.xtc -c membed_out.gro -e membed_out.edr (GROMACS v.4.5.6 / v.4.5.7)

The program will ask to select a group to embed in the membrane; select 1 for protein
Then the program will ask to select a group to embed Protein into; select 13 for DPPC lipid

Then, the program will run a 2 ps simulation during which it will resize the protein by a factor of 0.500 in the xy plane and 1.000 in the z direction (these can change) with respect to the geometrical center of all protein atoms that span the membrane region. At the same time, all lipid and water molecules that overlap with protein atoms will be removed.

g_membed tool will remove 18 DPPC and 45 water molecules that overlap with the protein. The exact number of each deleted component is displayed on the terminal session. Write down the exact number of each deleted component because you will need this information when you have to update the topology file.

5) Adding ions

Edit system.top correcting the number of DPPC lipids (512-18=494) and water molecules (44278-45=44233) left after the embedding of the protein (g_membed process). The file system.top should look then like this,

#include "gromos53a6_lipid.ff/forcefield.itp"
#include "protein.itp"
#include "dppc.itp"
#include "gromos53a6_lipid.ff/spc.itp"
#include "gromos53a6_lipid.ff/ions.itp"

[system]
; Name
AQP1 + DPPC + water

[molecules]
; Compound #mols
Protein_chain_A 1
DPP **494**
SOL **44233**

Prepare a .tpr file for the genion program. This program replaces solvent molecules by monoatomic ions at the position of the first atoms with the most favorable electrostatic potential or at random. The potential is calculated after every ion insertion using particle-based methods. For this calculation the user can specify a reaction field, shift function in the run input file but according to the developers the calculation of the potential is not reliable.

grompp -c membed_out.gro -p system.top -f min.mdp -o genion.tpr

Run the genion program to add counter ions. The total charge of the protein is +1 so we need 1 CL ion to obtain an electrically neutral system:

genion -s genion.tpr -p system.top -nn 1 -o system.gro
Select group 15 (SOL) when prompt to

In this way, 1 randomly selected water molecule is replaced with 1 CL ion.

Update system topology. Add the line #include "gromos53a6_lipid.ff/ions.itp" so that the final version of the system.top looks like:

#include "gromos53a6_lipid.ff/forcefield.itp"
#include "protein.itp"
#include "dppc.itp"
#include "gromos53a6_lipid.ff/spc.itp"
#include "gromos53a6_lipid.ff/ions.itp"

[system]
; Name
AQP1 + DPPC + water **+ ions**

[molecules]
; Compound #mols
Protein_chain_A 1
DPP 494
SOL 44232
CL **1**

genion tool reduces automatically the number of water molecules (SOL) by 1; from 44233 to 44232.

6) Energy minimization

Prepare the .tpr file for a 2000 step Steepest Descent energy minimization on the final solvated system that contains the ions,

grompp -f min.mdp -c system.gro -p system.top -o minim_in.tpr

Run the energy minimization,

mdrun -s minim_in.tpr -deffnm em_system_out

Make sure that the E_{pot} is negative and on the order of 10^5-10^6, as well as that the maxim force is no greater than 1000 $kJ^{-1}nm^{-1}$.

7) Short MD simulation

For the MD simulation of your system you can proceed (and is highly recommended) with the newest version of GROMACS that has several improved features regarding the parallelization of the MD simulations. In this tutorial, we run the simulations with GROMACS v.4.5.5.

Create an index file (index.ndx) that contains an additional group of atoms called SOL_CL using the program make_ndx,

make_ndx -f em_system_out.gro
At the prompt type
16 | 20
and then press q

this will create the SOL_CL group.

A file is generated, called index.ndx containing the atom numbers from all the default GROMACS atom groups (Protein, DPP, SOL etc) plus the SOL_CL group we created. You need this extra group as water and ions will be

temperature coupled as a whole. There are insufficient degrees of freedom in an ion-only group to justify a separate temperature coupling group.

- Equilibrate membrane and solvent around protein

Prepare a tpr file for a 200 ps simulation of the system applying harmonic positional restraints to all non-hydrogen protein atoms. The inclusion of restraints is achieved by the -DPOSRES flag in equil1.mdp file which tells GROMACS to take into account the following statement at the end of the protein topology file (protein.itp) #include "posre.itp".

The parameters for this simulation are in equil1.mdp. Spend some time, go through it and try to understand what does each MD simulation option do consulting the GROMACS manual (manual.gromacs.org/online/md_opt.html).

grompp -f equil1.mdp -c em_system_out.gro -p system.top -n index.ndx -o equil1.tpr

Run the simulation,

mdrun -s equil1.tpr -deffnm equil1_out

- Run unbiased MD simulation

Prepare a tpr file for a 500 ps simulation of the system without including the positional restraints for the protein. The parameters for this simulation are in equil2.mdp. We will run the unbiased simulation starting (coordinates, velocities) from the last step of the equilibration step

grompp -c equil1_out.gro -t equil1_out.cpt -p system.top -f md.mdp -n index.ndx -o md.tpr

Run the simulation,

mdrun -s md.tpr -deffnm md_out

For visualization of the simulation on VMD, remove first the periodicity from the trajectory (md_out.xtc) with trjconv command,

trjconv -s md.tpr -f md_out.xtc -o md_pbc.xtc -pbc mol -ur compact

and then load md_pbc.xtc into md_out.gro.

8) Loading structures/trajectories on VMD

After starting up VMD,

- Go to the File menu and select New Molecule.
- Determine the file type you want to open (gro, pdb etc) and browse for it by clicking Browse.
- Select the file you want to open from the file window and then click Load.
- From the "Graphics" menu select "Representations". In the "Selected Atoms" bar, you can define atom

selection you want to visualize. Click on the Selections tab to see the list of available selection keywords. E.g. to visualize the protein, simply type “protein” (without the quotes). Only the proteins atoms will appear then on the VMD window drawn in lines. You can select a different atom representation from the Drawing Method menu. To visualize an additional component of your system (e.g. the lipid), click on the Create Rep button in Graphical Representations. This will replicate the existing representation, the protein. In the Selected Atoms bar, type “resname DPP” and the lipid bilayer drawn in lines will appear on the VMD window. Resname is the keyword to select by residue name, SER, PHE, etc. If you know the residue number you want to visualize, you type “resid 86” in the Selected Atoms bar (after clicking on the Create Rep button).

- To visualize a trajectory you do the following. First, you load the structure used as input for the grompp command that created the tpr file of the simulation you want to visualize. For example, if you want equil2_pbc.xtc (final trajectory of the tutorial), you load on VMD the equil1_out.gro file. After you select the representation of the system, as described above, go to the File menu and select New Molecule. In the bar named Load Files for: change New Molecule to equil1_out.gro. Then in the Determine file type bar select the option “GROMACS XTC Compressed Trajectory”, click on Browse and select equil2_pbc.xtc from the file window. Click on Load to load the trajectory. To play it again, use the animation controls found at the bottom of the main VMD form. Some introductory tutorials on VMD are here and a full user guide is here.