Tutorial 6. SIRAH force field in GROMACS

Simulation of coarse grained membrane proteins in explicit solvent

By Exequiel Barrera
Mail any comment or suggestion to *spantano@pasteur.edu.uy*

This tutorial shows how to use the SIRAH force field to perform a coarse grained (CG) simulation of a protein embedded in a lipid bilayer using explicit solvent (called WatFour, WT4). The protocol consists of 5 simple steps: 1 download; 2 map; 3 insert the protein in the membrane; 4 solvate and; 5 run. The main references for this tutorial are: Machado et al. *SIRAH 2.0* [JCTC, **2019**, 15:2719], Barrera et al. *SIRAH Lipids* [JCTC, **2019**, 15:5674], Machado et al. *SIRAH Tools* [Bioinformatics, **2017**, 32:1568]. We strongly advise you to read these articles and complete Tutorials 3 and 5 before starting.

Required Software
GROMACS 4.6.7 or later version properly installed in your computer. The molecular visualization program VMD (freely available at *www.ks.uiuc.edu/Research/vmd*). The plotting software Grace (*plasma-gate.weizmann.ac.il/Grace*) and the R statistical package (*www.r-project.org*).

Prior knowledge
How to perform a standard atomistic molecular dynamic simulation with GROMACS. Have performed Tutorials 3 and 5 of SIRAH in GROMACS.

Hands on
0) Download the file *sirah_[version].gmx.tgz* from *www.sirahff.com* and uncompress it into your working directory. Notice: *[version]* should be replaced with the actual package version e.g.: x2_18-09

```
tar -xzvf sirah_[version].gmx.tgz
```

You will get a folder *sirah_[version].ff/* containing the force field definition, the SIRAH Tools in *sirah_[version].ff/tools/*, molecular structures to build up systems in *sirah_[version].ff/PDB/*, frequently asked questions in *sirah_[version].ff/tutorial/SIRAH_FAQs.pdf* and the required material to perform the tutorial in *sirah_[version].ff/tutorial/6/* .

Make a new folder for this tutorial in your working directory:

```
mkdir tutorial6; cd tutorial6
```

Create the following symbolic links in the folder *tutorial6*:

```
ln -s ../sirah_[version].ff sirah.ff
ln -s sirah.ff/residuetypes.dat
ln -s sirah.ff/specbond.dat
```

Notice: Files *residuetypes.dat* and *specbond.dat* are essential for the correct definition of molecular groups and auto-detection of disulfide bonds and cyclic DNA polymers.

1) Map the protonated atomistic structure of protein 2KYV to its CG representation:

```
./sirah.ff/tools/CGCONV/cgconv.pl\
 -i sirah.ff/tutorial/6/2kyv.pqr\
 -o 2kyv_cg.pdb
```

Notice: Mapping to CG requires the correct protonation state of each residue at a given pH. See *SIRAH_FAQs.pdf* and Tutorial 3 for cautions while preparing and mapping atomistic proteins to SIRAH.

Notice: If you already have an atomistic protein within a membrane, then you can simply map the entire system to SIRAH (this is highly recommended) and skip the step 2. By default no mapping is applied to lipids, as there is no standard naming convention for them. So users are requested to append a MAP file from the list in Table 1, by setting the flag *-a* in *cgconv.pl*. We recommend using PACKMOL (free at *http://m3g.iqm.unicamp.br/packmol*) for building the system. Reference building-block structures are provided at folder *sirah.ff/PDB/*, which agree with the mapping scheme in *sirah.ff/tools/CGCONV/maps/tieleman_lipid.map*. See *SIRAH_FAQs.pdf* for cautions on mapping lipids to SIRAH and tips on using fragment-based topologies.

Notice: You can access the help of the script *cgconv.pl* by executing:

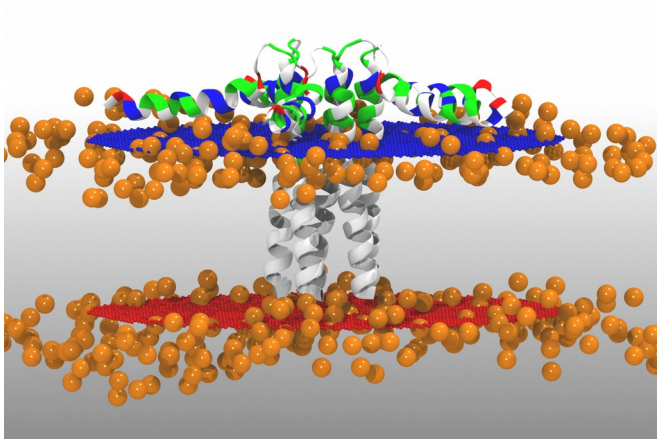
```
./sirah.ff/tools/CGCONV/cgconv.pl -h
```

2) Embed the protein in a lipid bilayer:
We will show one possible way to do it by starting from a pre-equilibrated CG membrane patch.

In *sirah.ff/tutorial/6/* you will find a pre-stabilized CG DMPC bilayer patch, concatenate it with the previously generated CG representation of the phospholamban (PLN) pentamer

```
head -qn -1 2kyv_cg.pdb ./sirah.ff/tutorial/6/DMPC_cg.pdb > 2kyv_DMPC_cg_init.pdb
```

Luckily, we already oriented the protein inside the membrane. For setting up your own system you can go to Orientations of Proteins in Membranes (OPM) database (*http://opm.phar.umich.edu*) and (if your structure is available) use the dummy atoms provided there to make them match with your membrane model (see right image).



Use VMD to delete lipid molecules in close contact with the protein:

For a proper treatment and visualization of the system in VMD you must first create a .gro file using the pdb2gmx command:

```
pdb2gmx\
 -f 2kyv_DMPC_cg_init.pdb\
 -o 2kyv_DMPC_cg_init.gro\
 -p init_topol\
 -i init_posre
```

When prompted, choose "*SIRAH force field*" and then "*SIRAH solvent models*".

Notice: GROMACS' commands in 5.x and later versions start with "*gmx*".

CG molecules are not recognized by molecular visualizers and will not display correctly. To fix this problem you may generate a PSF file of the system using the script *g_top2psf.pl*:

```
./sirah.ff/tools/g_top2psf.pl -i init_topol.top -o 2kyv_DMPC_cg_init.psf
```

Notice: This is the basic usage of the script *g_top2psf.pl*, you can learn other capabilities from its help:

```
./sirah.ff/tools/g_top2psf.pl -h
```

Open the files on VMD:

```
vmd 2kyv_DMPC_cg_init.psf 2kyv_DMPC_cg_init.gro -e sirah.ff/tools/sirah_vmdtk.tcl
```

Notice: VMD assigns default radius to unknown atom types, the script *sirah_vmdtk.tcl* sets the right ones. It also provides a kit of useful selection macros, coloring methods, backmapping utility and a command to calculate and display the secondary structure of SIRAH proteins. Use the command *sirah_help* in the Tcl/Tk console of VMD to access the manual pages.

In the VMD main window, select *Graphics* > *Representations*. In the *Selected Atoms* text entry type:

```
not (same residue as (sirah_membrane within 3.5 of sirah_protein))
```

Save the refined protein-membrane system: In the VMD main window click on `2kyv_DMPC_cg_init.psf`, then select *File* > *Save Coordinates.* In the Selected atoms option choose the selection you have just created and *Save* as `2kyv_DMPC_cg.pdb`.

From now on it is just normal GROMACS stuff!

3) Use pdb2gmx to convert your PDB file into GROMACS format:

```
pdb2gmx -f 2kyv_DMPC_cg.pdb -o 2kyv_DMPC_cg.gro
```

When prompted, choose "*SIRAH force field*" and then "*SIRAH solvent models*".

Notice: By default charged termini are used but it is possible to set them neutral with option *-ter*

Notice: Getting warning messages of long bonds is fine and expected due to the CG nature of the residue topologies. However missing atom messages are errors which probably trace back to the mapping step. In that case, check your atomistic and mapped structures and do not carry on the simulation until the problem is solved.

4) Solvate the system

Define the simulation box of the system:

```
editconf -f 2kyv_DMPC_cg.gro -o 2kyv_DMPC_cg_box.gro -box 12 12 12 -c
```

Add WT4 molecules:

```
genbox -cp 2kyv_DMPC_cg_box.gro -cs sirah.ff/wt416.gro -o 2kyv_DMPC_cg_solv1.gro
```

Notice: in GROMACS 5.x and later versions the command genbox was renamed to "*gmx solvate*"

Edit the *[molecules]* section in *topol.top* to include the number of added WT4 molecules:

Hint! If you forget to read the number of added WT4 molecules from the output of genbox, then use the following command line to get it

```
grep -c WP1 2kyv_DMPC_cg_solv1.gro
```

Topology before editing	Topology after editing

`[ molecules ]`
`; Compound #mols`
`Protein_chain_A 1`
`Protein_chain_B 1`
`Protein_chain_C 1`
`Protein_chain_D 1`
`Protein_chain_E 1`
`Lipid_chain_F 1`
	`[ molecules ]`
	`; Compound #mols`
	`Protein_chain_A 1`
	`Protein_chain_B 1`
	`Protein_chain_C 1`
	`Protein_chain_D 1`
	`Protein_chain_E 1`
	`Lipid_chain_F 1`
	`WT4 3875`

Notice: The number of added WT4 molecules (3875) may change according to the software version.

Remove misplaced WT4 molecules inside the bilayer or too close to the protein (read *sirah.ff/0ISSUES* and *SIRAH_FAQs.pdf* for comments on known solvation problems):

```
grompp\
 -f sirah.ff/tutorial/6/GPU/em1_CGLIPROT.mdp\
 -p topol.top -c 2kyv_DMPC_cg_solv1.gro -o 2kyv_DMPC_cg_solv1.tpr

echo "q" | make_ndx -f 2kyv_DMPC_cg_solv1.gro -o 2kyv_DMPC_cg_solv1.ndx

g_select\
 -f 2kyv_DMPC_cg_solv1.gro\
 -s 2kyv_DMPC_cg_solv1.tpr\
 -n 2kyv_DMPC_cg_solv1.ndx\
 -on rm_close_wt4.ndx\
 -sf sirah.ff/tutorial/6/rm_close_wt4.dat

editconf -f 2kyv_DMPC_cg_solv1.gro -o 2kyv_DMPC_cg_solv2.gro -n rm_close_wt4.ndx
```

Notice: New GROMACS versions may complain about the non-neutral charge of the system, aborting the generation of the TPR file by command *grompp*. We will neutralize the system later, so to overcame this issue, just allow a warning message by adding the following keyword to the *grompp* command line: *-maxwarn 1*

Edit the *[molecules]* section in *topol.top* to correct the number of WT4 molecules:

Hint! Use the following command to get the remaining WT4 molecules in the system.

```
grep -c WP1 2kyv_DMPC_cg_solv2.gro
```

Add CG counterions and 0.15M NaCl:

```
grompp\
 -f sirah.ff/tutorial/6/GPU/em1_CGLIPROT.mdp\
 -p topol.top\
 -c 2kyv_DMPC_cg_solv2.gro\
 -o 2kyv_DMPC_cg_solv2.tpr

genion\
 -s 2kyv_DMPC_cg_solv2.tpr\
 -o 2kyv_DMPC_cg_ion.gro\
 -np  95 -pname NaW\
 -nn 110 -nname ClW
```

When prompted, choose to substitute WT4 molecules by ions.

Notice: The available ionic species in SIRAH force field are: Na+ (NaW), K+ (KW) and Cl- (ClW). One ion pair (e.g. NaW-ClW) each 34 WT4 molecules renders a salt concentration of ~0.15M (see Appendix 1). Counterions were added according to Machado et al. *SPLIT* [JCTC, **2020**].

Edit the *[molecules]* section in *topol.top* to include the CG ions and the correct number of WT4.

Before running the simulation it may be a good idea to visualize your molecular system:

```
./sirah.ff/tools/g_top2psf.pl -i topol.top -o 2kyv_DMPC_cg_ion.psf
```

Use VMD to check how the CG system looks like:

```
vmd 2kyv_DMPC_cg_ion.psf 2kyv_DMPC_cg_ion.gro -e sirah.ff/tools/sirah_vmdtk.tcl
```

5) Generate position restraint files

To achieve a proper interaction between the protein and the bilayer, we will perform a equilibration step applying restraints over the protein backbone and lipids' phosphate groups.

Important! GROMACS enumerates atoms starting from 1 in each topology file (e.i. *topol_*.itp*). In order to avoid atom index discordance between the .gro file of the system and each topology, we will need independent .gro files for the PLN monomer and the DMPC bilayer.

Create an index file of the system with a group for a PLN monomer, then generate the .gro file of it. Notice: By default, WT4 and CG ions (NaW, ClW) are set to the group "*SIRAH-Solvent*" while DMPC (named CMM at CG level) is assigned to group "*Lipid*".

```
echo -e "ri 1-52\nq" | make_ndx -f 2kyv_DMPC_cg_ion.gro -o 2kyv_DMPC_cg_ion.ndx

editconf -f 2kyv_DMPC_cg_ion.gro -n 2kyv_DMPC_cg_ion.ndx -o 2kyv_cg_monomer.gro
```

When prompted, choose "*r_1-52"*.

Create an index file for the monomer and add a group for the backbone *GO* and *GN* beads.

```
echo -e "a GO GN \nq" | make_ndx -f 2kyv_cg_monomer.gro -o 2kyv_cg_monomer.ndx
```

Create a position restraint file for the monomer backbone

```
genrestr\
 -f 2kyv_cg_monomer.gro\
 -n 2kyv_cg_monomer.ndx\
 -fc 100 100 100\
 -o posre_BB.itp
```

When prompted, choose "*GO_GN*".

Edit each *topol_Protein_chain_*.itp* (A to E) to include the new position restraints:

Topology before editing	Topology after editing

`; Include Position restraint file`
`#ifdef POSRES`
`#include "posre_Protein.itp"`
`#endif`
	`; Include Position restraint file`
	`#ifdef POSRES`
	`#include "posre_Protein.itp"`
	`#endif`
	
	`#ifdef POSREBB`
	`#include "posre_BB.itp"`
	`#endif`

Use a similar procedure to set the positional restraints on lipid's phosphates.

```
editconf -f 2kyv_DMPC_cg_ion.gro -n 2kyv_DMPC_cg_ion.ndx -o DMPC_cg.gro
```

When prompted, choose "*Lipid*".

Create an index file of the membrane and add a group for phosphates (*BFO* beads).

```
echo -e "a BFO \nq" | make_ndx -f DMPC_cg.gro -o DMPC_cg.ndx
```

Create a position restraint file for phosphate groups in *z* coordinate.

```
genrestr -f DMPC_cg.gro -n DMPC_cg.ndx -fc 0 0 100 -o posre_Pz.itp
```

When prompted, choose "*BFO*".

Edit *topol_Lipid_chain_F.itp* to include the new position restraints and define the flag POSREZ to switch on these restraints in the input file (*.mdp*).

Topology before editing	Topology after editing

`; Include Position restraint file`
`#ifdef POSRES`
`#include "posre_Lipid_chain_F.itp"`
`#endif`
	`; Include Position restraint file`
	`#ifdef POSRES`
	`#include "posre_Lipid_chain_F.itp"`
	`#endif`
	
	`#ifdef POSREZ`
	`#include "posre_Pz.itp"`
	`#endif`

6) Run the simulation

The folder *sirah.ff/tutorial/6/GPU/* contains typical input files for energy minimization (*em_CGLIPROT.mdp*), equilibration (*eq_CGLIPROT.mdp*) and production (*md_CGLIPROT.mdp*) runs. Please check carefully the input flags therein.

Make a new folder for the run:

```
mkdir run; cd run
```

Energy Minimization of side chains by restraining the backbone::

```
grompp\
 -f ../sirah.ff/tutorial/6/GPU/em1_CGLIPROT.mdp\
 -p ../topol.top\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -c ../2kyv_DMPC_cg_ion.gro\
 -r ../2kyv_DMPC_cg_ion.gro\
 -o    2kyv_DMPC_cg_em1.tpr

mdrun -deffnm 2kyv_DMPC_cg_em1 &> EM1.log &
```

Energy Minimization of the whole system:

```
grompp\
 -f ../sirah.ff/tutorial/6/GPU/em2_CGLIPROT.mdp\
 -p ../topol.top\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -c    2kyv_DMPC_cg_em1.gro\
 -r    2kyv_DMPC_cg_em1.gro\
 -o    2kyv_DMPC_cg_em2.tpr

mdrun -deffnm 2kyv_DMPC_cg_em2 &> EM2.log &
```

Equilibration 1:
Position restraints are defined in *eq_CGLIPROT.mdp* file for protein backbone in xyz and phosphate groups (BFO beads) in *z* coordinate by setting keywords -DPOSREBB and -DPOSREZ, respectively.

```
grompp\
 -f ../sirah.ff/tutorial/6/GPU/eq1_CGLIPROT.mdp\
 -p ../topol.top\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -c    2kyv_DMPC_cg_em2.gro\
 -r    2kyv_DMPC_cg_em2.gro\
 -o    2kyv_DMPC_cg_eq1.tpr

mdrun -deffnm 2kyv_DMPC_cg_eq1 &> EQ1.log &
```

Equilibration 2:

```
grompp\
 -f ../sirah.ff/tutorial/6/GPU/eq2_CGLIPROT.mdp\
 -p ../topol.top\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -c    2kyv_DMPC_cg_eq1.gro\
 -o    2kyv_DMPC_cg_eq2.tpr

mdrun -deffnm 2kyv_DMPC_cg_eq2 &> EQ2.log &
```

Production (1μs):

```
grompp\
 -f ../sirah.ff/tutorial/6/GPU/md_CGLIPROT.mdp\
 -p ../topol.top\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -c    2kyv_DMPC_cg_eq1.gro\
 -o    2kyv_DMPC_cg_md.tpr

mdrun -deffnm 2kyv_DMPC_cg_md &> MD.log &
```

That’s it! Now you can analyze the trajectory.

Example of trajectory analysis

Process the output trajectory at folder run/ to account for the Periodic Boundary Conditions (PBC):

```
trjconv\
 -s    2kyv_DMPC_cg_em1.tpr\
 -n ../2kyv_DMPC_cg_ion.ndx\
 -f    2kyv_DMPC_cg_md.xtc\
 -o    2kyv_DMPC_cg_md_pbc.xtc\
 -pbc mol
```

When prompted, choose “System” for output.

Now you can check the simulation using VMD:

```
vmd ../2kyv_DMPC_cg_ion.psf ../2kyv_DMPC_cg_ion.gro 2kyv_DMPC_cg_md_pbc.xtc\
 -e ../sirah.ff/tools/sirah_vmdtk.tcl
```

Mapping atomistic lipids to SIRAH

Table 1. Available mapping files (MAPs) at folder *sirah.ff/tools/CGCONV/maps/* for converting atomistic lipid structures to SIRAH models. Important! MAPs can not inter-convert different name conventions, e.g. *amber_lipid.map* won't generate fragment-based residues from residue-based force fields. Due to possible nomenclature conflicts, users are advised to check and modify the MAPs as required.

MAP	Topol[1]	Compatibility	Source
amber_lipid.map	F	AMBER Lipid11-17 force fields	AMBER HTMD
GAFF_lipid.map	R	AMBER GAFF force field	LipidBook
charmm_lipid.map	R	CHARMM 27/36 force field, and "CHARMM compatible" GAFF nomenclature	CHARMM-GUI GROMACS LipidBook MemBuilder HTMD VMD
slipids.map	R	Stockholm lipids force field	SLIPIDS MemBuilder
OPLSA-AA_2014_lipid.map	R	All-atoms lipids for OPLS force field	Maciejewski et al. 2014
OPLSA-UA_lipid.map	R	United-atom lipids for OPLS force field	LipidBook
GROMOS43a1_lipid.map	R	United-atom lipids for GROMOS 43a1 and CKP force fields	LipidBook MemBuilder
GROMOS43a1-s3_lipid.map	R	United-atom lipids for GROMOS 43a1-s3 force field	GROMACS repo LipidBook MemBuilder
GROMOS53a6_lipid.map	R	United-atom lipids for GROMOS 53a6 force field	GROMACS repo LipidBook MemBuilder
tieleman_lipid.map	R	Berger lipids as implemented by Tieleman et al. for GROMOS force fields.	Tieleman LipidBook

[1] Fragment-based (F) or Residue-based (R) topology.

Appendix 1: Calculating ionic concentrations

$\rho_{WT4} = \rho_{H2O} = 1000$ g/L
$MW_{H2O} = 18$ g/mol
1 WT4 ~ 11 H_2O

$$M = \frac{mol}{V} \quad ; \quad n = mol\, N_A \quad ; \quad \rho = \frac{m}{V} \quad ; \quad m = mol\, MW$$

$$V = \frac{m}{\rho} = \frac{mol\, MW_{H_2O}}{\rho} = \frac{n_{H_2O} MW_{H_2O}}{N_A \rho} \quad ; \quad M = \frac{mol}{V} = \frac{n_{ion}}{N_A V} = \frac{n_{ion}}{N_A} \frac{N_A \rho}{n_{H_2O} MW_{H_2O}} = \frac{n_{ion} 1000}{n_{WT4}(11)(18)} \sim 5 \frac{n_{ion}}{n_{WT4}}$$

Number of WT4 molecules per ion at 0.15M: $n_{WT4} = 5 \frac{n_{ion}}{M} = \frac{5(1)}{0.15} \sim 34$