Tutorial

Switching Gō-Martini for Investigating Protein Conformational Transitions and Protein-Lipid Interactions

Introduction to Switching Gō-Martini

The Gō-like Martini model is an innovative fusion of the Gō-like model and the Martini model, which was first introduced by Proma et al.(2017). In this model, protein structures are maintained by the contact map interactions (rCSU and OV) using Lennard-Jones potentials. In contrast, the contact interactions in the traditional Martini model are depicted as harmonic potentials. The most notable distinction between these two potential functions is the energy when deviating from their equilibrium values. The Lennard-Jones potential approaches zero for extended interaction lengths, while the harmonic potential yields exceedingly high energies under similar conditions. This fundamental contrast endows the Gō-like Martini model with the capability of capturing more dynamics of protein conformations, including the folding and unfolding process of proteins, akin to other Gō-like models. However, the single-basin energy landscape constructed by the contact map in the Gō-like Martini model restricts the applicability of this model in exploring the conformational transitions between distinct states in multiple-state proteins. To address this limitation, here we will introduce a novel method, named **Switching Gō-Martini**, which is based on the switch approach and the Gō-like Martini model. Our method can not only efficiently capture the conformational transitions of proteins, but also reveal the associated proteins-lipids interactions.

Tutorial

In this section, we will utilize β2AR, a representative of GPCR, as an example to illustrate the capability of the Switching Gō-Martini method in sampling conformational transition pathways. In brief, we first generate two Gō-Martini models for active and inactive β2AR following the Martini tutorial. Subsequently, we perform a sequential simulation, running both systems one after the other, with a short-time relaxation process incorporated to relax the transition. For convenience, all the necessary files used within this tutorial (including worked files) are supplied.

(1) Preprocess PDB files (PDB code: 3SN6 and 2RH1) for the active and inactive β2AR. Water and other small molecules, such as the activator and inhibitor, should be removed. The missing loops and mutations in 3SN6 should be repaired by Modeller. And the missing side chains of 2RH1 should be added. Lastly, residues of proteins should be trimmed to have the same length. In this case, we reserve residues 30-230 and 265-341 of β2AR.

(2) Generate the Gō-Martini model for the active β2AR. We utilize martinize2 to generate the Martini topology of 3SN6 with flags to assign the Gō-like model.

```
martinize2 -f 3SN6_clean.pdb -o system.top -x Protein_Active_cg.pdb -dssp d
ssp -p backbone -ff martini3001 -govs-include -govs-moltype Active -cys aut
o -scfix
```

We also need to upload the atomistic pdb file to web-server and generate the contact map with default settings. Download and uncompress the generated .tgz file. Then, we use the script create_goVirt_for_multimer.py modified based on create_goVirt.py to generate all additional files. This script can be used for monomers or multimers. CG structure of the protein in pdb format (-s), the number of CG beads in the protein excluding the virtual Gō beads (–Natoms), the contact map file (-f), the reference atomistic structure in pdb format (-r), the prefix of the generated files (–moltype), and importantly, the dissociation energy of the Lennard-Jones potentials (–go_eps) are needed by this script.

```
python create_goVirt_for_multimer.py -r 3SN6_clean.pdb -s Protein_Active_c
g.pdb -f 3SN6_clean.map --moltype Active --go_eps 12 --Natoms 681
```

Then, we download the martini 3.0 force field and add statements (#include) to the file followed by renaming the file to martini_v3.0.0_Active.itp. Move these .itp files into a new directory for better management.

```
# download the martini_v3.0.0.itp
sed -i "s/\\[ nonbond_params \\]/\\#ifdef GO_VIRT\\n\\#include \\"BB-part-d
ef_VirtGoSites.itp\\"\\n\\#endif\\n\\n\\[ nonbond_params \\]/" martini_v3.
0.0.itp
echo -e "\\n#ifdef GO_VIRT \\n#include \\"go-table_VirtGoSites.itp\\"\\n#en
dif" >> martini_v3.0.0.itp
mv martini_v3.0.0.itp martini_v3.0.0_Active.itp
mkdir ActiveITP
mv *.itp ActiveITP
```

(3) Generate the Gō-Martini model for the inactive β2AR as we just did for the active state of β2AR.

```
martinize2 -f 2RH1_clean.pdb -o system.top -x Protein_Inactive_cg.pdb -dssp
dssp -p backbone -ff martini3001 -govs-include -govs-moltype Inactive -cys
auto -scfix

# upload 2RH1_clean.pdb to the web-server and download the contact map.
python create_goVirt_for_multimer.py -r 2RH1_clean.pdb -s Protein_Inactive_
cg.pdb -f 2RH1_clean.map --moltype Inactive --go_eps 12 --Natoms 681

# download the martini_v3.0.0.itp again.
sed -i "s/\\[ nonbond_params \\]/\\#ifdef GO_VIRT\\n\\#include \\"BB-part-d
ef_VirtGoSites.itp\\"\\n\\#endif\\n\\n\\[ nonbond_params \\]/" martini_v3.
0.0.itp
echo -e "\\n#ifdef GO_VIRT \\n#include \\"go-table_VirtGoSites.itp\\"\\n#en
dif" >> martini_v3.0.0.itp
mv martini_v3.0.0.itp martini_v3.0.0_Inactive.itp
mkdir InactiveITP
mv *.itp InactiveITP
```

(4) Insert the CG protein of the active β2AR into a POPC membrane and solvate the system with water and ions by using the script insane.py. Be aware that there are issues with ion names and

counts for the added ions that require manual correction.

```
wget http://www.cgmartini.nl/images/tools/insane/insane.py
python2 insane.py -f Protein_Active_cg.pdb -box 8,8,10 -l POPC -o ions.gro
-salt 0.15 -charge auto -center -sol W  2>system.top

# Repair the wrong ion names and ion counts.
sed -i "s/NA+    NA+/NA      NA/g" ions.gro
sed -i "s/CL-    CL-/CL      CL/g" ions.gro
sed -i "s/NA+/NA /g" system.top
sed -i "s/CL-/CL /g" system.top
vi system.top # Delete 2 CL in system.top and ions.gro. The error is becaus
e the virtual atoms without charges are also calculated by insane.py.
vi ions.gro
```

(5) Prepare the following files: system_Active.top, system_Inactive.top, .ndx index file (containing Protein, Membrane, Solvent), and the .mdp files (em.mdp, npt.mdp, md.mdp, and md_relax.mdp). Particularly, the md_relax.mdp is important, in which a short-time relaxation is conducted for 1000 steps with a time step of 0.002 ps to smooth the transition and reduce the risk of system crashes.

```
cp system.top system_Active.top
vi system_Active.top # add #define GO_VIRT and #include the topol files (.i
tp) we need.
cp system_Active.top system_Inactive.top
sed -i 's/Active/Inactive/g' system_Inactive.top

gmx make_ndx -f ions.gro -o index.ndx # Protein, Membrane and Solvent
```

If everything goes well, our next step is to run the simulations as follows.

```
# Minimization
gmx grompp -f em.mdp -c ions.gro -p system_Active.top -o em.tpr -n index.nd
x
gmx mdrun -v -deffnm em

# Equilibration
gmx grompp -f npt.mdp -o npt.tpr -c em.gro -r em.gro -p system_Active.top -
n index.ndx
gmx mdrun -v -deffnm npt

# Production
gmx grompp -f md.mdp -o md1.tpr -c npt.gro -p system_Active.top -n index.nd
x
gmx mdrun -deffnm md1 -v

gmx grompp -f md_relax.mdp -o md2_1.tpr -c md1.gro -p system_Inactive.top -
n index.ndx
gmx mdrun -deffnm md2 -v -s md2_1.tpr
```

```
gmx grompp -f md.mdp -o md2.tpr -c md1.gro -p system_Inactive.top -n index.
ndx
gmx mdrun -deffnm md2 -v  -cpi md2.cpt -s md2.tpr -append
```

(6) Finally, analyze the simulations we just obtained. The first thing we should do is to use trjconv to post-process the trajectories by concatenating them, correcting periodicity, and removing protein translation and rotation. Then, utilize VMD for simulation viewing and analysis as needed.

```
gmx trjcat -f md[1-2].xtc -o md_whole.xtc -settime -dt 100 << EOF
c
c
EOF
echo 1 0 |gmx trjconv -s md1.tpr -f md_whole.xtc -o md_nojump.xtc -pbc noju
mp -center
echo 1 0 |gmx trjconv -s md1.tpr -f md_nojump.xtc -o md_mol.xtc -pbc mol -c
enter
echo 1 0 |gmx trjconv -s md1.tpr -f md_mol.xtc -o md_rottrans.xtc -fit rot+
trans
echo 0 | gmx trjconv -s md1.tpr -f md_rottrans.xtc -o md_start.pdb -dump 0
```