Tutorial

Prerequisites

We provide here a very brief tutorial of how to setup a simulation system using our new parameters starting from PDB files. We use as an example the KcsA system investigated in our work. The following software and files are required to finish the tutorial.

- The MD python library, namely MDAnalysis, is needed. For details of the installation of MDAnalysis, please visit https://www.mdanalysis.org/.
- 3f7v.pdb: KcsA potassium channel open crystal structure.
- 1k4c.pdb: KcsA potassium channel closed crystal structure.
- hinge.py, insert_param.py and get_ENs provided in this package.

Procedure of generating topology files for KcsA

1) All the topology files needed for our model are built on standard MARTINI topology files generated using martini.py script (download available here) or CHARMM-GUI MARTINI MAKER (http://www.charmm-gui.org/?doc=input/martini). These files include Protein_A.itp, Protein_B.itp, Protein_C.itp and Protein_D.itp.

   To use martini.py to construct the topology of KcsA, please enter the following command:

python martini.py -f 3f7v.pdb -o system.top –x cg.pdb -dssp dssp -p backbone -ff martini22

2) Next, we create a file containing additional parameter set needed for modeling each individual TM helix. For example, TM1 helix of chain A of KcsA starts from residue 30 and ends at residue 52, to get such a file for this TM helix, we can enter the command as follows:

python hinge.py 3f7v.pdb Protein_A.itp A 30 52 > chainA_TM1_param.itp

where chainA_TM1_param.itp contains all the needed parameters for the targeted TM helix. This script also generates another file called *hinge.xvg* which contains the bending angles of each residue in this helix which will be used to calculate the force constants of the dihedral restraints according to Eq. 4 in our paper.

Parameter sets for all other TM helices in the same chain of KcsA can be obtained in the same way. For convenience, all the commands of doing these tasks are combined in a text called README which can be run all at once.

We then combine the new parameter sets of all TM helices (TM1, P-helix and TM2) of chain A into a single file as follows:

cat chainA_TM1_param.itp chainA_P-helix_param.itp chainA_TM2_param.itp > chainA_param.itp

3) Next, we need to incorporate the new parameter set for chain A into the standard topology file for MARTINI as follows:

   python insert_param.py Protein_A.itp chainA_param.itp

   Once you execute the script, a file new_Protein_A.itp containing all the needed parameters for all the TM helices of chain A of KcsA are will be generated.

   Similarly, you should be able to obtain the combined topology files for all the other subunits of KcsA including new_Protein_B.itp, new_Protein_C.itp and new_Protein_D.itp.

4) Finally, make the following changes to the overall topology of the system that are specified in system.top :

   Replacing

   #include “Protein_A.itp”

   #include “Protein_B.itp”

   #include “Protein_C.itp”

   #include “Protein_D.itp”

   with

   #include “new_Protein_A.itp”

   #include “new_Protein_B.itp”

   #include “new_Protein_C.itp”

   #include “new_Protein_D.itp”.

Procedure of generating elastic network (EN) bonds for KcsA

Thus far, we have already obtained the parameters that can be used to simulate any membrane proteins with hinge structures. Yet, as illustrated in our paper, in order to accurately model conformational change of membrane proteins resulting from hinging motions, a modified elastic network may be needed. In the following part, we will use KcsA as an example to show how such an elastic network can be incorporated. Because of multiple chains, there are elastic restraints that need to be applied between different chains. To implement these restraints, all the 4 topology files for each chain, namely new_protein_A/B/C/D.itp, need to be combined into a single topology.

1) For the reason above, a script called topology_combinate.py was employed for this task:

   python topology.py 4 new_Protein_A.itp new_Protein_B.itp new_Protein_C.itp new_Protein_D.itp

   The Protein_total.itp topology file generated has the same section of topologies of different chains combined and assigns global indices to all the atoms of KcsA.

2) Next, we need to provide distance restraints between CG sites used for the elastic network model. Again, the site indices should be global. To this end, we first assign all chains of KscA a single chain ID, say ‘A’ in a modified PDB file, namely 3f7v-mod.pdb. Then we run the following Martini.py again.

python martini.py -f 3f7v-mod.pdb -o system.top –x cg.pdb -dssp dssp -p backbone -ff elnedyn22

By running the command above, we can obtain a file called Protein_A.itp which contains a list of distance restraints for an open structure of KcsA. We name it as open_Protein_A.itp. Then, we can obtain the corresponding file (closed_Protein_A.itp) for a closed KcsA in the same way:

python martini.py -f 1k4c.pdb -o system.top –x cg.pdb -dssp dssp -p backbone -ff elnedyn22

3) To obtain appropriate EN bonds for KcsA, we removed all the local constraints within TM helices as the structures of these parts are managed by our new parameters. In addition, we compare the open and closed structures of KcsA and select the pairs whose distance varies by over 10% between the two structures. All these tasks will be finished by a single script as follows:

python get_ENs.py open_Protein_A.itp closed_Protein_A.itp > new_ENs.xvg

4) Finally, the content in new_ENs.xvg generated as above is inserted into the global topology from step 1 right after section [bonds].

Note: In step 2, in order to obtain the one-to-one correspondence between the open and closed topology files, the total chain length of open and closed crystal structures must be identical for both structures. In addition, if the helical disruptions are assigned as a TM helix by martini.py

(with a type ‘T’), you need to manually change the types of these residues from “T” back to “H” in open/closed_Protein_A.itp obtained in step 1 before proceeding to step 3.

For example, after replacing turn structural type in disrupted residue 53, you should see the tip file change from

```
105    N0    50   ALA    BB   105   0.0000 ; 2
106    Na    51   ASP    BB   106   0.0000 ; 2
107    Qa    51   ASP   SC1   107  -1.0000 ; 2
108    Na    52   LEU    BB   108   0.0000 ; 2
109    C1    52   LEU   SC1   109   0.0000 ; 2
110   Nda    53   VAL    BB   110   0.0000 ; T
111    C2    53   VAL   SC1   111   0.0000 ; T
112   Nda    54   MET    BB   112   0.0000 ; 3
113    C5    54   MET   SC1   113   0.0000 ; 3
114   Nda    55   GLY    BB   114   0.0000 ; 3
115   Nda    56   LEU    BB   115   0.0000 ; 3
116    C1    56   LEU   SC1   116   0.0000 ; 3
```

to

```
105    N0    50   ALA    BB   105   0.0000 ; 2
106    Na    51   ASP    BB   106   0.0000 ; 2
107    Qa    51   ASP   SC1   107  -1.0000 ; 2
108    Na    52   LEU    BB   108   0.0000 ; 2
109    C1    52   LEU   SC1   109   0.0000 ; 2
110   Nda    53   VAL    BB   110   0.0000 ; H
111    C2    53   VAL   SC1   111   0.0000 ; H
112   Nda    54   MET    BB   112   0.0000 ; 3
113    C5    54   MET   SC1   113   0.0000 ; 3
114   Nda    55   GLY    BB   114   0.0000 ; 3
115   Nda    56   LEU    BB   115   0.0000 ; 3
116    C1    56   LEU   SC1   116   0.0000 ; 3
```

In step 4, additional constraints added to the N-terminal of TM1 of KcsA each subunit for maintaining their secondary structure. These restraints are as follows:

```
  2     5       1    0.56602 RUBBER_FC*1.000000
  3     7       1    0.54622 RUBBER_FC*1.000000
180   183       1    0.56608 RUBBER_FC*1.000000
181   185       1    0.54624 RUBBER_FC*1.000000
358   361       1    0.56599 RUBBER_FC*1.000000
359   363       1    0.54620 RUBBER_FC*1.000000
536   539       1    0.56606 RUBBER_FC*1.000000
537   541       1    0.54625 RUBBER_FC*1.000000
```

Building and running MARTINI simulation

We can use these new topology files to build and run simulation. The procedures are same as in standard MARTINI, please visit MARTINI website (http://cgmartini.nl/index.php) or CHARMM-GUI MARTINI MAKER (http://www.charmm-gui.org/?doc=input/martini).