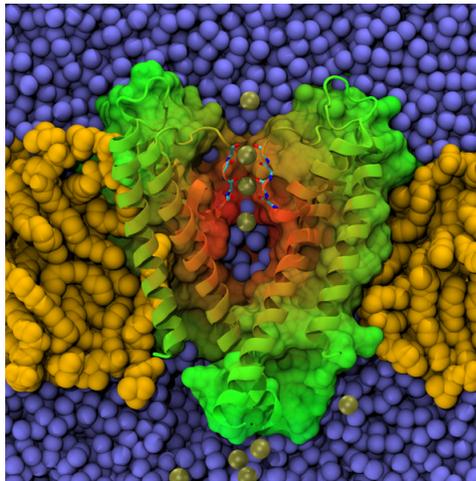


University of Illinois at Urbana-Champaign
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group
Computational Biophysics Workshop

Membrane Proteins Tutorial



Alek Aksimentiev

Marcos Sotomayor

David Wells

Current editor:

Paween Mahinthichaichan

November 2016

A current version of this tutorial is available at
<http://www.ks.uiuc.edu/Training/Tutorials/>
Join the `tutorial-1@ks.uiuc.edu` mailing list for additional help.

Contents

1	Building a Structural Model of KcsA	6
1.1	Downloading and Viewing the Protein	6
1.2	Examining the PDB File	8
1.3	Building the KcsA Tetramer	9
1.4	Generating PSF and PDB Files for KcsA	12
1.5	Solvating the Protein (Optional)	18
2	Placing KcsA in a Membrane	24
2.1	Building a Membrane Patch	24
2.2	Alignment of Membrane and Protein	25
2.3	Combination of Membrane and Protein	26
2.4	Solvation and Ionization	29
3	Running a Simulation of KcsA	34
3.1	Melting of Lipid Tails	34
3.2	Minimization and Equilibration with Protein Constrained	39
3.3	Equilibration with Protein Released	41
3.4	Production Runs	43

Introduction

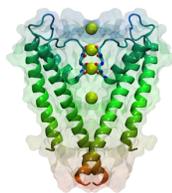
This tutorial is designed to guide users of VMD and NAMD through all the steps required to set up a membrane protein system for molecular dynamics simulations. The tutorial assumes that you already have a working knowledge of VMD and NAMD. For the accompanying VMD and NAMD tutorials go to: <http://www.ks.uiuc.edu/Training/Tutorials/>

This tutorial has been designed specifically for VMD 1.8.6 and NAMD 2.6 but will also work with VMD 1.8.7 and NAMD 2.7. It should take about 5 hours to complete in its entirety. This time can be reduced by skipping the optional section 1.5 and by using the example scripts where provided.

The tutorial is subdivided into three separate units. The first unit covers steps required to set up a structural model of a membrane protein starting from a raw PDB file. The second unit describes the steps needed to place the protein in a native-like membrane environment. Finally, the third unit describes the steps required to minimize and equilibrate the resulting system with NAMD.

The examples in the tutorial will focus on the study of the KcsA membrane protein – an archetypal potassium channel with very interesting properties. Throughout the text, some material will be presented in separate “boxes”. These boxes include complementary information to the tutorial, such as information about the biological role of KcsA, and tips or technical details that can be further explored by the advanced user.

If you have any questions or comments on this tutorial, please email the TCB Tutorial mailing list at tutorial-l@ks.uiuc.edu. The mailing list is archived at http://www.ks.uiuc.edu/Training/Tutorials/mailling_list/tutorial-l/.



KcsA. This tutorial will focus on preparing a system that contains the potassium channel KcsA embedded in a fully hydrated membrane. KcsA was the first ion channel crystallized; it is made of four identical subunits forming a tetramer and features a selectivity filter which permits highly selective conduction of potassium ions at nearly bulk diffusion rates across the membrane. KcsA belongs to a family of channels found in almost all organisms. These channels have diverse functions and have been implicated in osmotic regulation and neuronal signaling.

Required programs

The following programs are required for this tutorial:

- **VMD:** Available at <http://www.ks.uiuc.edu/Research/vmd/> (for all platforms)
- **NAMD:** Available at <http://www.ks.uiuc.edu/Research/namd/> (for all platforms)
- **Solvate (recommended):** Available at <http://www.mpibpc.mpg.de/grubmueller/solvate/> (be sure to add the following line to `solvate.c` before compilation: `#include <stdlib.h>`)

Getting Started

You can find the files for this tutorial in the `mem-tutorial-files` directory. Below you can see in Fig. 1 the files and directories of `mem-tutorial-files`.

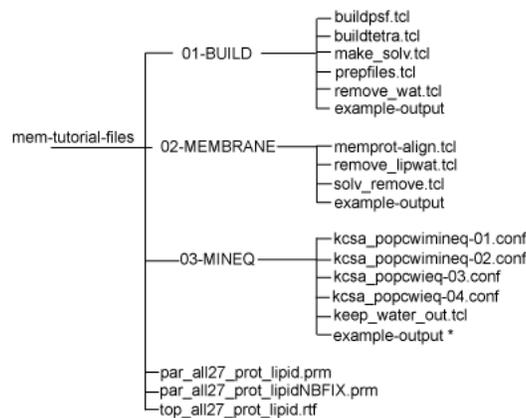


Figure 1: Directory structure of `mem-tutorial-files`. * This example-output directory is present only in the extended version of the tutorial.

To start VMD type `vmd` in a Unix terminal window, double-click on the VMD application icon likely located in the **Applications** folder in Mac OS X, or click on the **Start** → **Programs** → **VMD** menu item in Windows.

Acknowledgments

Development of this tutorial is supported in part by the National Institutes of Health (P41-RR005969 - Resource for Macromolecular Modeling and Bioinformatics) and the National Science Foundation (DMR-095595 and PHY-0822613).

1 Building a Structural Model of KcsA

In this unit you will build the KcsA tetramer solvated in water, learning how to take a raw protein structure and build a simulation-ready system out of it.

1.1 Downloading and Viewing the Protein

Our first step is to download the raw protein structure from the Protein Data Bank, an online repository for protein structures experimentally resolved, and view it using VMD. The PDB file `1K4C.pdb` contains the atom coordinates of a monomer of KcsA.

- 1 Open a web browser and navigate to `www.pdb.org`. The file we will download is also provided in `mem-tutorial-files/01-BUILD/example-output/`.
- 2 In the search box at the top center, select `PDB ID or keyword`, then type `1K4C` and click `Site Search`. Each PDB file in the database has a unique code of four letters or numbers identifying it. One may also search using the name of one of the publishing authors.
- 3 On the left-hand side, click `Download Files` below `1K4C`, then click `PDB text`. You will be prompted to save the file `1K4C.pdb`, which you should place in `mem-tutorial-files/01-BUILD/`.

From now on, in this unit we will work with files located at `mem-tutorial-files/01-BUILD`. Use the `cd` command in a terminal window (or in VMD's Tk Console when applicable) to set the correct directory.

- 4 Now close the web browser, open VMD, and load the file `1K4C.pdb` that you just saved.
- 5 This file contains not only a monomer of the KcsA protein, but also everything else solved from the crystal structure. Set up the following representations to see this:

Selection	Drawing Method	Coloring Method
<code>protein</code>	<code>NewCartoon</code>	<code>Chain</code>
<code>not protein</code>	<code>VDW</code>	<code>Name</code>

We see that the PDB file also contains water molecules, ions, large antibody complexes (used to crystallize the KcsA) as well as some lipid fragments (Fig. 2). These different structural elements are divided into "chains" in the PDB file, which we will see in detail in the next section. For now, the point is simply that the PDB files you download will contain structures that you want and structures that you don't, but that these are easily separated from one another.

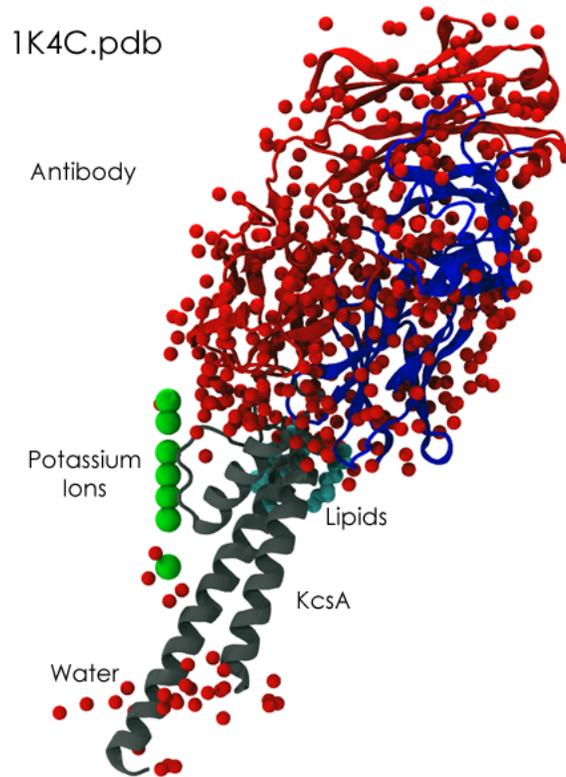


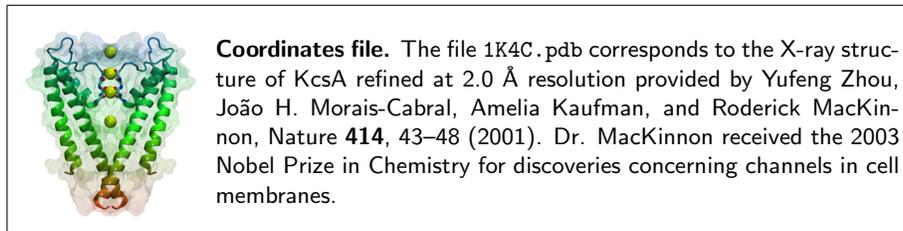
Figure 2: Molecules in file 1K4C.pdb

Now that the file is loaded, you can explore it by trying out different selections and drawing options, and use the Query mouse mode (Mouse menu, or shortcut 0) to see information about individual atoms. Get familiar with the structure and be sure that you have identified KcsA as the chain that contains α -helices. Try to identify the file contents mentioned above, and which chain each belongs to.

6 Once you are done exploring the structure, close VMD.



Webpdb. VMD can download a pdb file from the Protein Data Bank if a network connection is available. Just type the four letter code of the protein in the File Name text entry of the Molecule File Browser window and press the Load button. VMD will download it automatically.



1.2 Examining the PDB File

PDB files are simply text files, and contain more information than just atomic coordinates. Here we will look at the file you downloaded in a text editor and examine its contents.

- 1 Open the the file 1K4C.pdb in your favorite text editor, e.g. by typing `emacs 1K4C.pdb` in a terminal window.
- 2 First, simply look through the file. Notice the plethora of information: authors, resolution, experimental method, secondary structure, etc. The atomic coordinates don't start until line 577!
- 3 Now we will look at some specific parts which are particularly important. First, look at the COMPND items beginning on line 4.

```
COMPND    MOL_ID: 1;
COMPND    2 MOLECULE: ANTIBODY FAB FRAGMENT HEAVY CHAIN;
COMPND    3 CHAIN: A;
COMPND    4 MOL_ID: 2;
COMPND    5 MOLECULE: ANTIBODY FAB FRAGMENT LIGHT CHAIN;
COMPND    6 CHAIN: B;
COMPND    7 MOL_ID: 3;
COMPND    8 MOLECULE: POTASSIUM CHANNEL KCSA;
COMPND    9 CHAIN: C;
COMPND   10 FRAGMENT: POTASSIUM CHANNEL KCSA;
COMPND   11 ENGINEERED: YES;
COMPND   12 MUTATION: YES
```

These items list the protein chains which you looked at above (Fig. 2). We see that chain C is the KcsA channel we are interested in. In addition to those listed here, there is a chain X which contains the water molecules, ions, and lipid fragments.

- 4 We want to build a system with a whole KcsA channel, a tetramer, yet the PDB file we have downloaded contains only one monomer. To generate the missing subunits, we must apply transformation matrices. These matrices are given in REMARK 350 of the PDB file. Find this section.

```

REMARK 350 BIOMOLECULE: 1
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B, C
REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000          0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000          0.00000
REMARK 350   BIOMT3   1  0.000000  0.000000  1.000000          0.00000
REMARK 350   BIOMT1   2 -1.000000  0.000000  0.000000        310.66000
REMARK 350   BIOMT2   2  0.000000 -1.000000  0.000000        310.66000
REMARK 350   BIOMT3   2  0.000000  0.000000  1.000000          0.00000
REMARK 350   BIOMT1   3  0.000000 -1.000000  0.000000        310.66000
REMARK 350   BIOMT2   3  1.000000  0.000000  0.000000          0.00000
REMARK 350   BIOMT3   3  0.000000  0.000000  1.000000          0.00000
REMARK 350   BIOMT1   4  0.000000  1.000000  0.000000          0.00000
REMARK 350   BIOMT2   4 -1.000000  0.000000  0.000000        310.66000
REMARK 350   BIOMT3   4  0.000000  0.000000  1.000000          0.00000

```

5 Look at the lines labeled BIOMT n where n is a number. Each set of three BIOMT n lines describes the rotations and translations necessary to generate each subunit (notice that the first matrix is simply a unit matrix, corresponding to no transformation.) In the next section, you will see how to use these matrices to generate the complete tetramer.

6 Now scroll down a bit more, and look at the REMARK 465 and REMARK 470 lines. These lines describe residues and atoms missing from the structure, respectively. All structures are missing some atoms and residues and it is important to check whether missing parts are relevant for the question being addressed with the simulations. Below you will see how *psfgen* is able to guess the coordinates of some of the missing atoms.

7 Close the text editor.

1.3 Building the KcsA Tetramer

As you discover yourself, the PDB file `1K4C.pdb` contains only one subunit of the KcsA tetramer. However, for most proteins, there is only one subunit, or all subunits are present in the original PDB file. Here you will build the whole KcsA tetramer, noting that for many other membrane proteins, this step will not be necessary.

1 Start a new VMD session and select the Extensions \rightarrow Tk Console menu item in the VMD Main window. The next four steps are also replicated in the script `builddtetra.tcl`.

2 Create and save segment A of the protein by typing the following commands in the Tk Console window.

```

cd "< your home directory >/mem-tutorial-files/01-BUILD/"
mol new 1K4C.pdb
set all [atomselect top all]
$all set segname A
$all writepdb KCSA-A.pdb
$all delete

```

Note that you should always delete selections once you are finished with them.



Rotations and Translations in VMD. The transformation matrices provided in REMARK 350 of a PDB file (lines labeled BIOMT) contain 4 columns and 3 rows each. The matrix formed by the first three columns and three rows describes a rotation, while the last column describe a translation. VMD uses 4×4 matrices to perform rotations and translations, so one must add a fourth row ($\{0\ 0\ 0\ 1\}$) to the matrices provided in the PDB file.

Now you will transform the coordinates of segment A into those of segment B, C, and D by using the appropriate matrix transformations as described below. Note that we will not replicate potassium ions since they are located in the symmetry axis of the protein.

- 3 Create and save segment B of the protein by typing the following commands in the Tk Console window.

```

set sel [atomselect top "all and not name K"]
$sel set segname B
$sel move {{-1.0 0.0 0.0 310.66} {0.0 -1.0 0.0 310.66}
{0.0 0.0 1.0 0.0} {0.0 0.0 0.0 1.0}}
$sel writepdb KCSA-B.pdb
$sel delete

```

Note that in the previous step we used the second matrix (BIOMT2) of REMARK 350 in 1K4C.pdb with an additional row. Be sure to add spaces between vectors delimited by $\{\}$.

- 4 Delete the current molecule, load the KcsA crystal structure again and create segment C of the protein by typing the following commands in the Tk Console window.

```

mol delete top
mol new 1K4C.pdb
set sel [atomselect top "all and not name K"]
$sel set segname C
$sel move {{0.0 -1.0 0.0 310.66} {1.0 0.0 0.0 0.0}
{0.0 0.0 1.0 0.0} {0.0 0.0 0.0 1.0}}
$sel writepdb KCSA-C.pdb
$sel delete

```

- 5 Finally, repeat the same procedure to create segment D by typing the following commands:

```
mol delete top
mol new 1K4C.pdb
set sel [atomselect top "all and not name K"]
$sel set segname D
$sel move {{0.0 1.0 0.0 0.0} {-1.0 0.0 0.0 310.66}
{0.0 0.0 1.0 0.0} {0.0 0.0 0.0 1.0}}
$sel writepdb KCSA-D.pdb
$sel delete
```

- 6 Exit VMD and locate the four files named KCSA-A.pdb, KCSA-B.pdb, KCSA-C.pdb, and KCSA-D.pdb, each one containing one segment of the tetramer. If anything seems wrong, use the script `buildtetra.tcl` to regenerate the correct files.

Now that you have created all monomers you will merge them into one single file.

- 7 Use the `cat` command in Unix to concatenate all files by typing in a Terminal window: `cat KCSA-A.pdb KCSA-B.pdb KCSA-C.pdb KCSA-D.pdb > KCSA.pdb`. If you are using Windows, use a text editor to copy and paste the contents of all files into one file called KCSA.pdb.

- 8 Using a text editor open the file KCSA.pdb, search for `END` and delete the lines:

```
END
CRYST1 155.330 155.330 76.270 90.00 90.00 90.00 P 1 1
```

that are between segments A and B, B and C, as well as those found between segments C and D.

Directly editing pdb files (as done above) is undesirable, you may want to create a Tcl script that performs the same operations while preserving appropriate indexing and formatting instead.

- 9 Save the modified file as KCSA-ALL.pdb. Open VMD, load KCSA-ALL.pdb, and set up the following representations:

Selection	Drawing Method	Coloring Method
protein	NewCartoon	SegName
not protein	VDW	Name



Figure 3: Top and side views of the KcsA tetramer with an antibody complex (1K4C.pdb).

In VMD, you should be able to see the full, tetrameric, KcsA channel along with water, ions, lipid fragments, and large antibody complexes used to crystallize it (Fig 3).



Homework. Create a VMD script that automates the construction of multimeric structures. The script should be able to read the transformation matrices described in REMARK 350 from a raw PDB file and output the multimeric form of the protein in a single pdb file. Solution: check the mono2poly script available at http://www.ks.uiuc.edu/Research/vmd/script_library/scripts/



Aligning the protein to the z -axis. The KcsA channel structure studied here has its principal axis along the channel with the z -axis. This configuration is very convenient for embedding the protein in a membrane and also facilitates the use of several analysis tools. However, membrane proteins obtained from the Protein Data Bank are often not aligned along the z -axis. For those cases, VMD provides the *orient* plugin, which permits alignment of principal axes of a molecule to x, y, z , directions. The plugin can be found at: http://www.ks.uiuc.edu/Research/vmd/script_library/scripts/orient/ You may also want to check the “Orientations of Proteins in Membranes” database at <http://opm.phar.umich.edu/>

1.4 Generating PSF and PDB Files for KcsA

Now that you have built the whole structure for KcsA and its antibody complex, you will generate PSF and PDB files for just the channel protein, crystallographic water molecules, and ions. You will generate these files step by

step using VMD and psfgen. The first five steps are included in the script `prepfiles.tcl`.

- 1 Be sure that `KCSA-ALL.pdb` is loaded in VMD as the top molecule. Then, select the Extensions → Tk Console menu item in the VMD Main window.

- 2 Psfgen requires not covalently linked segments to be in separate PDB files. Save each KcsA segment by typing in the Tk Console window:

```
cd "< your home directory >/mem-tutorial-files/01-BUILD/"
foreach S { A B C D } {
  set seg [atomselect top "segname $S and chain C and protein"]
  $seg writepdb seg$$S.pdb
  $seg delete
}
```

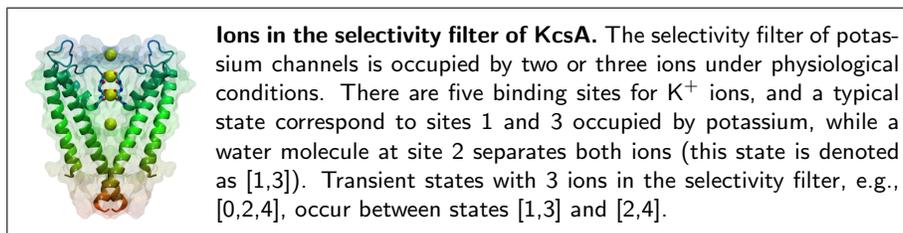
Check that the files `segA.pdb`, `segB.pdb`, `segC.pdb`, `segD.pdb`, were properly generated by looking at them with a text editor.

Each of the files you saved contains a single segment (monomer) of the KcsA membrane protein.

- 3 Now you will save a segment containing potassium ions by typing in the Tk Console window:

```
set pot [atomselect top "name K and resid 3001 3003 3005 3006"]
$pot set name POT
$pot set resname POT
$pot writepdb pot.pdb
```

Note that we have selected four out of seven possible potassium ions. We choose a configuration of ions that is actually feasible, with two ions in the selectivity filter and two in the intracellular and extracellular vestibules of the protein. We will use the rest of the ions to place water molecules in the selectivity filter.



- 4 Select the remaining potassium ions and transform them into water molecules by typing in the Tk Console window:

```

set npot [atomselect top "name K and resid 3002 3004 3007"]
$npot set name OH2
$npot set resname TIP3
$npot writepdb filtwat.pdb

```

It is always desirable to keep as much information from the crystal structure as possible. In this exercise we will not include in our model antibodies or lipid fragments resolved in the crystal, but we will keep crystallographic water molecules located nearby the channel and in protein cavities.

- 5 Select and save water molecules around the KcsA channel by typing in the Tk Console window:

```

foreach S {A B C D} {
  set wat [atomselect top "segname $S and
  resname HOH and within 10 of (chain C and protein)"]
  $wat writepdb crystwat$$S.pdb
  $wat delete
}

```

In case anything failed during the above steps, use the script `prefiles.tcl` to create again the corresponding files.



Water Molecules. Water molecules located in internal cavities of proteins often play a functionally relevant role. While these water molecules might already be present in the crystal structure being used, you may want to use specific software to determine if additional water molecules can be placed around or inside your protein. The program DOWSER has been specifically designed to place water molecules in internal cavities, and keeps or discards them based on energetics. In addition to adding water molecules, it can also evaluate crystallographic waters such as those seen in 1K4C.pdb above.

Now you should have separate PDB files for each protein segment, for ions, and for selected crystallographic water molecules. However, before generating PSF and PDB files for the KcsA tetramer, we also need to check whether disulfide bonds between cysteine residues exist and what the protonation states of charged residues and histidines are.

- 6 In VMD, which should be displaying the `KCSA-ALL.pdb` molecule, create the following additional graphical representations:

- 7 By looking at the above representations in the VMD OpenGL display you should be able to identify one cysteine residue per monomer (Cys90).

Selection	Drawing Method	Coloring Method
protein and chain C	NewCartoon	ColorID 6
chain C and resname CYS	VDW	Name
resname HIS and segname A and chain C	VDW	Name
chain C and segname A and protein and within 5 of resname HIS	Licorice	Name

Since these residues are far away from each other, we can conclude that there are no disulfide bonds present in the KcsA structure (Fig. 4 A). This may not be the case for other proteins.

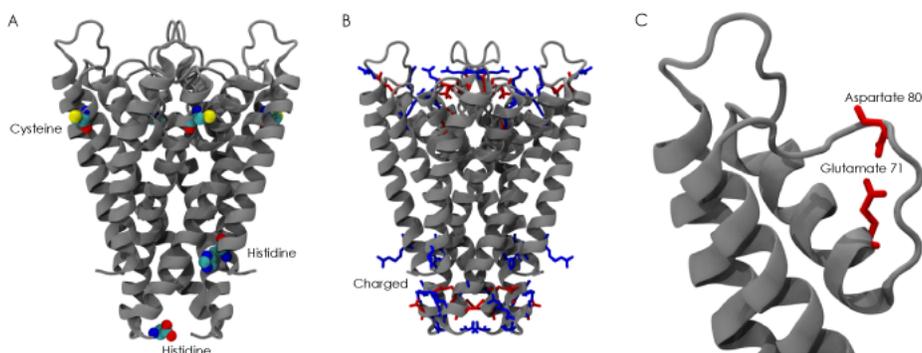


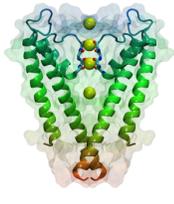
Figure 4: Inspecting the KcsA structure. (A) Cysteine and histidine residues. No evident disulfide bonds or histidine protonation states can be observed. (B) Charged residue. Notice how charged residues are distributed in solvent accessible areas. (C) Interaction between residues Glutamate 71 (Glu71) and Aspartate 80 (Asp80). One of these residues must share a proton with the other.

Usually, visual inspection permits one to identify whether the side chain of a histidine residue is involved in hydrogen bonding with surrounding molecules. If that's the case, one can then decide whether the δ nitrogen of histidine is protonated (residue name "HSD"), the ϵ nitrogen is protonated ("HSE"), or both nitrogens are protonated ("HSP").

8 Identify the histidines present in segment A.

The KcsA structure provided in 1K4C.pdb features two histidines (His25 and His124). His124 lacks the side chain, and no obvious protonation state can be

visualized for residue His25 (Fig. 4 A). Thus we will model both as HSE residues.



pH- and voltage- dependent gating of KcsA. Identifying the correct protonation state of histidines might be essential in proteins such as KcsA. This potassium channel has been shown to be activated by proton binding in the intracellular side of the channel. In addition, although KcsA lacks a canonical voltage-sensing domain, KcsA gating is influenced by voltage. The voltage and pH sensors are likely different physical entities (Julio F. Cordero-Morales, Luis G. Cuello, and Eduardo Perozo, *Nature Structural & Molecular Biology* **13**, 319–322, 2006).

9 Create the following additional graphical representation in VMD:

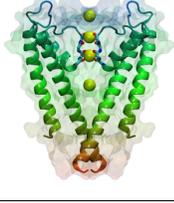
Selection	Drawing Method	Coloring Method
charged and chain C	Licorice	ResType

10 Identify all charged residues. Look at residues Arg27, Arg52, Arg64, Arg89, Arg117, Arg121, and Arg122. Note that Arg117 is missing its side chain. All arginines are located in regions that are likely solvated (Fig. 4 B). Thus we will model them as charged.

11 Look at residues Glu51, Glu71, Asp80, Glu118, and Glu120. Do you notice something unusual?

Residues Asp80 and Glu71 are very close to each other and actually interact through a water molecule. Likely, both residues share one proton and are not negatively charged at the same time (Fig. 4 C). Classical molecular dynamics simulations cannot accurately describe such state. Therefore, we will model Glu71 as a neutral amino-acid by applying the patch GLUP, as indicated below.

12 Exit VMD.



Protonation states of ionizable residues. It is usually desirable to use advanced pK_a calculations to reliably assign protonation states. The ionization state of residues such as glutamate, aspartate, lysine, and arginine can be extremely relevant for the function of a protein. For instance, the protonation state of Glu71 has been shown to be relevant for the dynamics and stability of KcsA (Simon Bernèche and Benoît Roux, *Biophysical Journal* **82**, 772–780, 2002).

You should have by now ten (!) files (`segA.pdb`, `segB.pdb`, `segC.pdb`, `segD.pdb`, `pot.pdb`, `filtwat.pdb`, and `crystwatA.pdb`, `crystwatB.pdb`, `crystwatC.pdb`,

crystwatD.pdb. We will use all these files to generate the corresponding PSF and PDB files with VMD, psfgen and the `buildpsf.tcl` script provided in `memtut-files/01-BUILD/`. The script already takes into account the information you gather about the protonation state of ionizable residues.

- 13** Open the file `buildpsf.tcl` with a text editor. The content should look like:

```
package require psfgen
topology ../top_all127_prot_lipid.rtf
pdbalias residue HIS HSE
pdbalias atom ILE CD1 CD
pdbalias atom HOH O OH2
pdbalias residue HOH TIP3
foreach S { A B C D } {
  segment $S {
    pdb seg$$S.pdb
  }
  coordpdb seg$$S.pdb $S
  patch GLUP $$S:71
  regenerate angles dihedrals
  segment WC$$S {
    auto none
    pdb crystwat$$S.pdb
  }
  coordpdb crystwat$$S.pdb WC$$S
}
segment I {
  pdb pot.pdb
}
coordpdb pot.pdb I
segment WF {
  auto none
  pdb filtwat.pdb
}
coordpdb filtwat.pdb WF
guesscoord
writepdb kcsav.pdb
writepsf kcsav.psf
exit
```

- 14** Read the script carefully and make sure you understand each command line. Check how we selected the protonation state of histidines and how we used a patch to protonate Glu71.

The script builds each segment of the protein along with the crystallographic water molecules we selected, the water molecules we placed in the selectivity

filter, and potassium ions.

- 15 Execute the script by typing in a Unix terminal `vmd -dispdev text -e buildpsf.tcl > buildpsf.rep` (if you are using Mac OS X or Windows, type in the Tk Console window `source buildpsf.tcl` instead).
- 16 Analyze the standard output of the script by opening the file `buildpsf.rep` in a text editor.

Most of the problems when building a protein model occur in the previous step. Warning messages in `buildpsf.rep` can usually be ignored, but be sure you understand the reason for the warning.

- 17 Use VMD to visualize and check the files `kcsav.psf` and `kcsav.pdb` generated in the previous step.

Now you have PSF and PDB files for the KcsA tetramer with ions and some water molecules.



Using *AutoPSF*. VMD provides a plugin called *AutoPSF* which automates the steps described above for the generation of PSF and PDB files. While it is good that you familiarize with the details of each step in the structure building process, *AutoPSF* can save you substantial time. *AutoPSF* can be launched by selecting the Extensions → Modeling → Automatic PSF Builder menu item. Use it only if you fully understand what you are doing.



Challenge. Using an approach similar to that used for the construction of the KcsA tetramer and its PSF and PDB files, try now to build the crystallographic unit cell containing 8 crystallographically related molecules. Look at REMARK 290 in the `1K4C.pdb` file.

1.5 Solvating the Protein (Optional)

The next step is to solvate the protein. To do this, we use Helmut Grubmüller's SOLVATE program. This program will fill in any empty space inside the pore, as well as surround it with water. We must then delete most of it, as there must be no water between the protein and the lipid bilayer we will embed it in later (note that Grubmüller's SOLVATE is a stand alone software, different than the Solvate package provided within VMD). While recommended, in the interest of time, this section is optional.

- 1 First, enter `solvate` in a terminal window to see some information about the program and to check that it is properly installed.

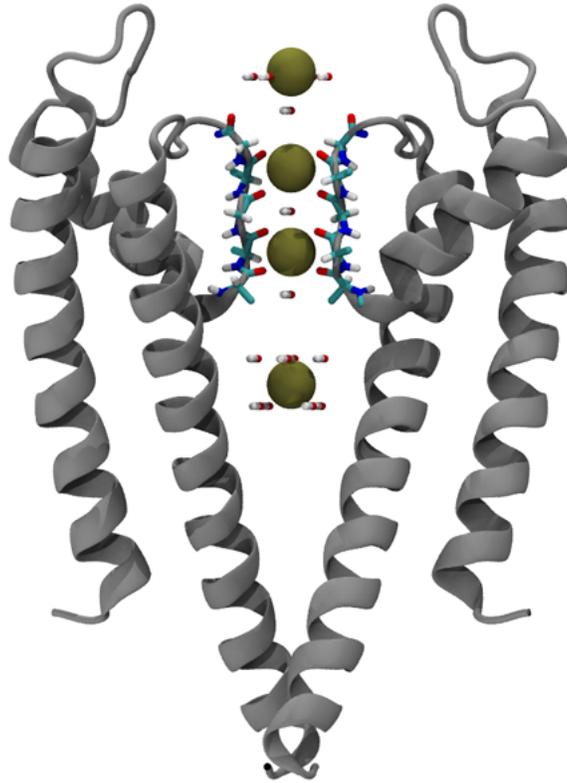


Figure 5: Snapshot of KcsA's selectivity filter obtained using files `kcsav.psf` and `kcsav.pdb`. Note the presence of hydrogen atoms and selected potassium ions, as well as crystallographic water molecules.

2 Now we will run Solvate on the PDB file you just produced. Type the following command in a terminal window:

```
solvate -t 3 -n 8 -w kcsav solkcsa
```

The options provided have the following meanings:

Option	Meaning
-t 3	Make the water layer 3 Å thick
-n 8	Use 8 gaussians for solvent boundary
-w	Output water molecules only

Also notice that we must leave the suffix `.pdb` off of the file names. This program may take a few minutes to run.

- 3 Now we will combine the output PDB file `solkcsa.pdb` with the rest of the system. Do this by entering:

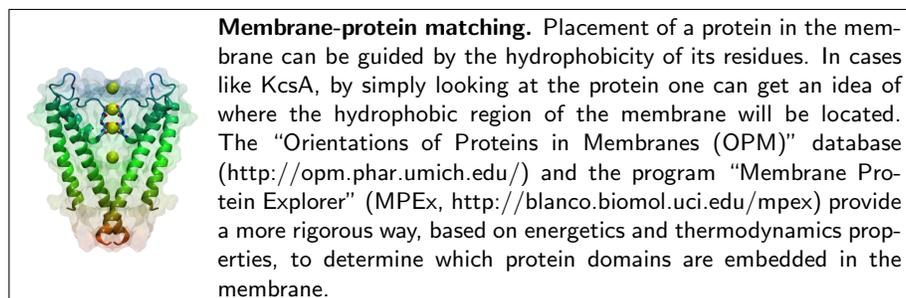
```
vmd -dispdev text -e make_solv.tcl
(or type in the Tk Console source make_solv.tcl).
```

The content of the script is:

```
set sol_infile solkcsa.pdb
set kcsa_inbase kcsav
set outbase kcsav_solv_raw
package require psfgen
resetpsf
topology ../top_all127_prot_lipid.rtf
segment SOLV {
  auto none
  first NONE
  last NONE
  pdb $sol_infile
}
coordpdb $sol_infile SOLV
readpsf ${kcsa_inbase}.psf
coordpdb ${kcsa_inbase}.pdb
writepdb ${outbase}.pdb
writepsf ${outbase}.psf
exit
```

This creates the corresponding segment for the new water molecules, combining them with your previous psf and pdb files.

Now we will remove unwanted water molecules, which basically means finding good selection criteria in VMD. Here, we want to eliminate water molecules located in the hydrophobic protein-membrane interface (Fig. 6).



- 4 To make selection easier, we will first center everything on the origin. Open VMD and be sure that you are in the `mem-tutorial-files/01-BUILD/` directory.

- 5 Enter the following commands in the Tk Console window:

```
mol new kcsav_solv_raw.psf
mol addfile kcsav_solv_raw.pdb
set all [atomselect top all]
$all moveby [vecinvert [measure center $all]]
display resetview
```

The `measure center` command finds the geometric center of the selection you give it. It can also find a weighted center, such as center of mass, center of charge, etc. The result of the command is a vector, which we invert using the `vecinvert` command, and this is in turn passed to the `moveby` option of the selection. Finally, we reset the view so we can see the centered molecule again.

- 6 In VMD, create the following graphical representations:

Selection	Drawing Method	Coloring Method
<code>protein</code>	VDW	ResType
<code>water</code>	Lines	ColorID 0
<code>water and z<-20</code>	Lines	ColorID 1
<code>water and z>10</code>	Lines	ColorID 1

Protein residues are now colored based on their hydrophobicity and acidity. Charged residues are colored blue and red, green residues are hydrophilic, and white residues are hydrophobic. The hydrophobic part of the protein will be interacting with the membrane. Can you guess then the location of the membrane? We will eliminate water molecules located in the hydrophobic region of the protein (Fig. 6).

- 7 Change the representation involving water molecules so as to cover different regions along the z -axis. Is the original selection of water molecules reasonable? Does it leave space for the membrane?
- 8 Now we will “mark” the water we don’t want to keep. For this, we will use the Beta field, marking “bad water” with a beta value of 1. Since most of the water will be deleted, we will mark all Solvate water, then change the ones we want to keep. Type in the Tk Console window:

```
set solv [atomselect top "segname SOLV"]
$solv set beta 1
```



The PDB B-factor field. The “B” field of a PDB file typically stores the “temperature factor” for a crystal structure and is read into VMD’s “Beta” field. Since we are not currently interested in this information, we can recycle this field to store our own numerical values and use them for different purposes.

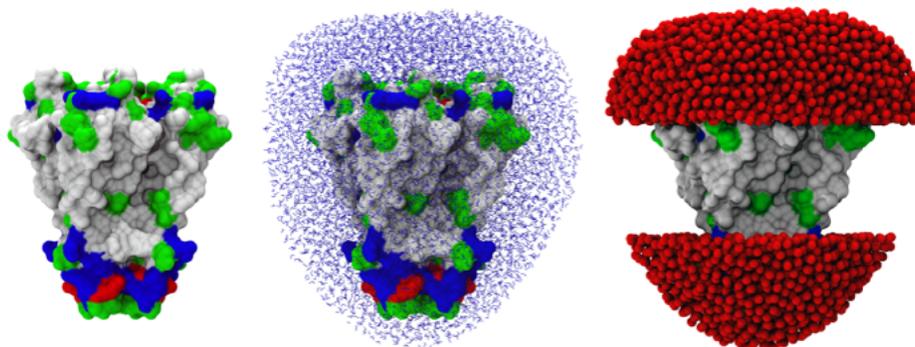


Figure 6: Solvating exposed areas of a membrane protein. The left panel shows KcsA colored by residue type. An hydrophobic belt can be identified. The center panel shows the outcome of raw solvation. The right panel depicts KcsA surrounded by water at its solvent-exposed sites (leaving an appropriate space for the membrane).

9 Now we set our selection text, and unmark the waters which it selects:

```
set seltext "segname SOLV and same residue as \
((z < -20) or (z>10))"
set sel [atomselect top $seltext]
$sel set beta 0
```

Notice how easily we can select atoms depending on their spatial location. Other geometric selections are equally easy. Also notice the use of **same residue as**: if any atom of the water molecule fits our geometric criteria, all atoms of that molecule are marked.

10 We will delete by **segid** and **resid**, since each water molecule is its own residue. Therefore, we now make a list of these for the marked molecules:

```
set badwater [atomselect top "name OH2 and beta > 0"]
set seglist [$badwater get segid]
set reslist [$badwater get resid]
```

Since we are deleting by residue, we only want one entry per molecule in each of these lists, and so we select all water oxygen atoms which are marked.

11 Now we use **psfgen** to read the PSF and PDB files, and delete the residues we just marked:

```
mol delete all
package require psfgen
resetpsf
topology ../top_all27_prot_lipid.rtf
readpsf kcsav_solv_raw.psf
coordpdb kcsav_solv_raw.pdb
foreach segid $seglist resid $reslist {
    delatom $segid $resid
}
writepdb kcsa_solv.pdb
writepsf kcsa_solv.psf
```

- 12 Check with VMD the files `kcsa_solv.pdb` and `kcsa_solv.psf`. If something went wrong, you may want to use the script `remove_wat.tcl` to create the corresponding files again.

We now have a complete protein with water molecules in just the right places (Fig. 6). Most of the hard work is done. In the next section, we will embed the protein system in a lipid bilayer, then solvate the entire system and set the proper ion concentration.

2 Placing KcsA in a Membrane

In this second unit, you will learn how to take your solvated KcsA channel and place it in a membrane. Then you will solvate and ionize the entire system.

2.1 Building a Membrane Patch

The first step is to prepare a complete membrane. This is usually done by replicating a pre-equilibrated patch of membrane and water, then trimming it if necessary. The VMD Membrane Builder plugin automates this process. The steps in this section and the following one (Alignment of Membrane and Protein) are also covered in the script `memprot-align.tcl`.

NOTE: If you skipped the previous section, you should either use the files `kcsa_solv.psf` and `kcsa_solv.pdb` from the example output, or replace every instance of these files with `kcsav.psf` and `kcsav.pdb` in the following instructions.

- 1 Navigate to the directory `mem-tutorial-files/02-MEMBRANE`, open VMD, and select the Extensions → Modeling → Membrane Builder menu item in the VMD Main window.
- 2 In the Membrane window, select the POPC Lipid and change the Membrane X Length and the Membrane Y Length to 80. Set the Output Prefix to `< your home directory >/mem-tutorial-files/02-MEMBRANE/popc`. Click the Generate Membrane button.

A patch of POPC membrane should appear in the VMD OpenGL display, along with the two corresponding files in your `02-MEMBRANE` directory.

- 3 Set up the following representations:

Selection	Drawing Method	Coloring Method
<code>resname POPC</code>	Lines	SegName
<code>water</code>	Lines	Name

Notice how different patches of membranes have been combined and how lipid head-groups are solvated (Fig. 7 A).



More membranes. In some cases certain types of lipids play key roles in the function of membrane proteins. Therefore, membrane proteins should be placed in their native-like environment, namely, in hydrated bilayers made of different combinations of lipids as found in *in vivo* conditions. VMD provides POPC and POPE membranes, but the CHARMM force-field supports several other types of lipids. Remember to set up the temperature of your simulation above the melting temperature of lipids used in your system.

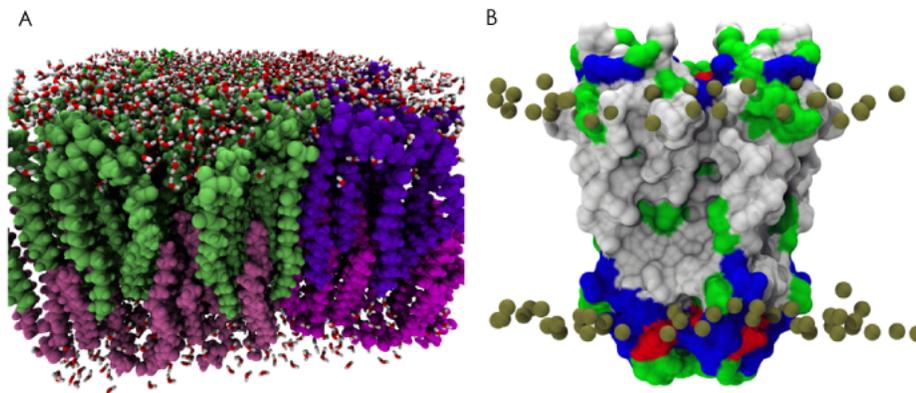


Figure 7: Embedding KcsA in a membrane patch. (A) POPC membrane patch. (B) KcsA protein and location of membrane headgroups.

2.2 Alignment of Membrane and Protein

Now that we have a suitable membrane patch and a partially solvated protein, the next step is to align them with one another properly.

- 1 Open the Tk Console window. Be sure your `popc.psf/pdb` files are loaded as the top molecule and set the current directory to `memtut-files/02-MEMBRANE`.
- 2 Create a selection for the membrane and load the set of files for KcsA with the following commands:


```
set popc [atomselect top all]
set kcsamol [mol new ../01-BUILD/kcsa_solv.psf]
mol addfile ../01-BUILD/kcsa_solv.pdb
set kcsa [atomselect $kcsamol all]
```
- 3 We will first align the membrane with its center of mass:


```
$popc moveby [vecinvert [measure center $popc weight mass]]
$popc writepdb popc.TEMP.pdb
```
- 4 For the channel, we will use the center of mass of the main center vestibule, i.e. residues 97–106. This choice should be tailored to the specific membrane protein being simulated. We will also rotate the channel about the z-axis, to align the roughly square top end with the x- and y-axes:


```
set vest [atomselect $kcsamol "protein and resid 97 to 106"]
$kcsa moveby [vecinvert [measure center $vest weight mass]]
display resetview
$kcsa move [transaxis z -25]
$kcsa writepdb kcsa.TEMP.pdb
```

Notice the use of the `transaxis` function. This is one of a few functions which generate transformation matrices of the type we took from the original PDB file. In this case, the matrix rotates about the z-axis by -25 degrees.

- 5 Now we simply combine the two temporary files into one set of PSF and PDB files:

```
mol delete all
package require psfgen
resetpsf
readpsf popc.psf
coordpdb popc_TEMP.pdb
readpsf ../01-BUILD/kcsa_solv.psf
coordpdb kcsa_TEMP.pdb
writepsf kcsa_popc_raw.psf
writepdb kcsa_popc_raw.pdb
file delete kcsa_TEMP.pdb
file delete popc_TEMP.pdb
```

- 6 Load the files `kcsa_popc_raw.psf` and `kcsa_popc_raw.pdb` in VMD. If something went wrong, you may want to regenerate the files by using the script `memprot-align.tcl`.
- 7 Before we move on, let's use one of VMD's many coloring methods to make sure our protein is lined up with the membrane in a natural way. To do this, set up the following representations:

Selection	Drawing Method	Coloring Method
protein	VDW	ResType
resname POPC and name P1	VDW	Name

The residues of the protein are now colored based on their hydrophobicity and acidity. We see that the headgroup phosphates (labeled P1 for POPC membranes) are aligned well with the tyrosine residues, colored green on the upper corners of the KcsA, as well as the arginine residues on the lower corners (Fig. 7 B). The white residues between the headgroups are hydrophobic, meaning we have a good alignment with the membrane. In some cases, it might be necessary to align protein and membrane manually.

Next, we will take out the lipids overlapping the KcsA.

2.3 Combination of Membrane and Protein

Now we need to make room for the KcsA in the membrane layer, so that the protein doesn't overlap any lipid molecules. As in the section above on using

Grubmüller's Solvate program, we want to find good selection criteria in VMD, and then mark the molecules we want to delete. All commands in this section are also included in the script `remove_lipwat.tcl`.

We will again use the beta field of the atoms to mark the “bad lipids”. This time, though, the selection will be slightly more complicated.

- 1 Enter the following commands to set beta of all atoms to zero:

```
mol delete all
mol new kcsa_popc_raw.psf
mol addfile kcsa_popc_raw.pdb
set POPC "resname POPC"
set all [atomselect top all]
$all set beta 0
```

Note that we also set a shortcut for the POPC selection `resname POPC`.

- 2 We will make three different selections, first marking lipids whose phosphorus atom lies well within the region of the KcsA, then marking remaining lipids within a certain distance of the protein:

```
set seltext1 "$POPC and same residue as \
(name P1 and z>0 and abs(x)<15 and abs(y)<15)"
set seltext2 "$POPC and same residue as \
(name P1 and z<0 and abs(x)<10 and abs(y)<10)"
set seltext3 "$POPC and same residue as (within 0.6 of protein)"
set sel1 [atomselect top $seltext1]
set sel2 [atomselect top $seltext2]
set sel3 [atomselect top $seltext3]
$sel1 set beta 1
$sel2 set beta 1
$sel3 set beta 1
set badlipid [atomselect top "name P1 and beta > 0"]
set seglistlipid [$badlipid get segid]
set reslistlipid [$badlipid get resid]
```

We choose based on the phosphorus atom because there is only one of them per POPC lipid.

Note that for a cylindrical protein we could have used a geometric criteria like $x^2 + y^2 > 225$, but because of the rather conical shape of KcsA, we used two different selections for the two layer of lipids. In the present example, the third selection (`seltext3`) may override the previous two selections, but that may not be the case in proteins featuring pores.

- 3 Use VMD to look at the selections created and try to find a better selection criteria so as to keep as much non-overlapping lipids as possible around KcsA.
- 4 As we saw above, the membrane also came with some water molecules solvating its headgroups. We must remove the water molecules overlapping with the protein:

```

set seltext4 "(water and not segname WCA WCB WCC WCD WF SOLV) \
  and same residue as within 3 of \
  ((same residue as (name P1 and beta>0)) or protein)"
set seltext5 "segname SOLV and same residue as \
  within 3 of lipids"
set sel4 [atomselect top $seltext4]
set sel5 [atomselect top $seltext5]
$sel4 set beta 1
$sel5 set beta 1
set badwater [atomselect top "name OH2 and beta > 0"]
set seglistwater [$badwater get segid]
set reslistwater [$badwater get resid]

```

- 5 Now, we delete the atoms as we did before:

```

mol delete all
resetpsf
readpsf kcsa_popc_raw.psf
coordpdb kcsa_popc_raw.pdb
foreach segid $seglistlipid resid $reslistlipid {
  delatom $segid $resid
}
foreach segid $seglistwater resid $reslistwater {
  delatom $segid $resid
}
writepsf kcsa_popc.psf
writepdb kcsa_popc.pdb

```

- 6 Load the files `kcsa_popc.psf` and `kcsa_popc.pdb` into VMD and check that there is no overlap between the channel and the membrane (Fig. 8). As usual, if you detect something wrong, you may want to regenerate the files using the script `remove_lipwat.tcl`

We are almost there! We only have to solvate (with VMD this time) and ionize the system, then we can start using NAMD to actually perform a simulation of KcsA in its native-like environment.

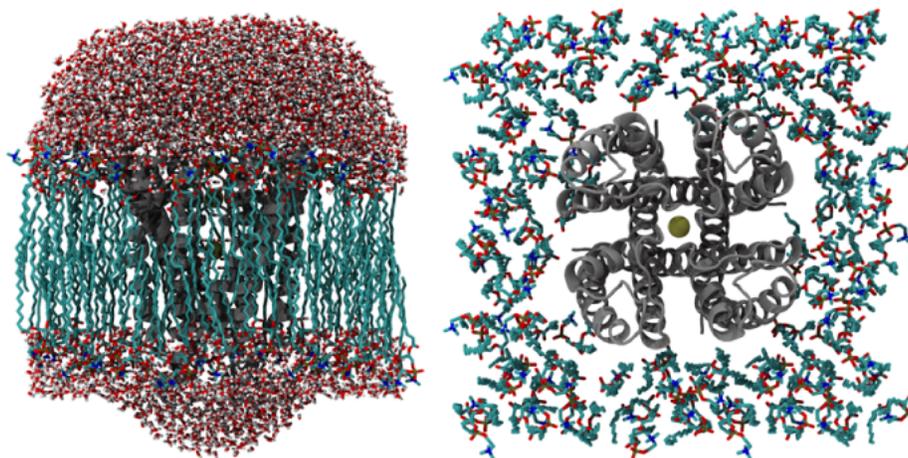


Figure 8: Top and side views of KcsA embedded in a membrane patch (obtained using files `kcsa_popc.psf` and `kcsa_popc.pdb`). Note the lack of overlap between the protein and the membrane.

2.4 Solvation and Ionization

We will now solvate and ionize the system. VMD's Solvate plugin is quite simple: it places your solute in a box of water of a specified size, then removes water which is within a certain cut-off distance. We will remove any water which adds inside the lipid bilayer as well. Steps 2-6 are also included in the script `solv_remove.tcl`.

- 1 First, let's use the `minmax` option of the `measure` command to see how big our water layer is:

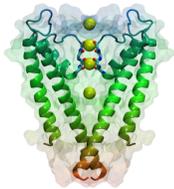
```
mol delete all
mol newf kcsa_popc.psf
mol addfile kcsa_popc.pdb
set water [atomselect top water]
measure minmax $water
```

You should get something like the following:

```
{-37.5699996948 -37.6370010376 -39.4020004272} \
{38.3590011597 38.8470001221 49.1450004578}
```

The size of the water box we make should be of similar size in the `xy`-plane. In fact, a water box slightly smaller in the `xy`-plane is desirable, since non-equilibrated membranes tend to shrink. In the `z`-direction, there is more choice. The size is often dictated by the distance to the periodic

image and the total number of atoms in your system: having a smaller water box is a cheap way of making a smaller system.



Size of simulation cell. Periodic boundary conditions are generally used to avoid surface effects and because its use permits the efficient computation of long-range electrostatic interactions. However, if the size of the simulation cell is too small, periodic images will interact with themselves affecting the outcome of the simulation. When choosing the size of your membrane patch and water box, be sure to leave at least 15 Å of space between the protein and the simulation cell boundaries on each direction. Heavily charged proteins may require even more spacing. One can also use the Debye length of the solution as a good criteria to determine the distance between periodic images. In any case, take into account that the example provided here is designed for users with limited access to computational resources (a larger membrane patch would be desirable).

- 2 We will make our system a nice cuboid, $\sim 2 \times 38$ Å on the plane of the membrane. Type the following commands to do this:

```
package require solvate
solvate kcsa_popc.psf kcsa_popc.pdb -o kcsa_popc_water_TEMP -b 1.5 \
-minmax {{-38 -38 -39} {39 39 50}}
```

The `-b 1.5` option tells `solvate` to remove atoms within 1.5 Å of the solute.

Now let's take a look at what `solvate` has produced.

- 3 Delete the `kcsa_popc` molecule and load, if not already loaded, the files `kcsa_popc_water_TEMP.pdb` and `kcsa_popc_water_TEMP.psf`.
- 4 Show only `segid WT1` to `WT99`. By default, the `segid` of the water produced by `solvate` begins with the prefix `WT`.

You can see that `solvate` put some water inside the lipid bilayer and around the protein (Fig. 9). We don't want this, so we must once again remove some atoms.

- 5 Type in the Tk Console window:

```
set all [atomselect top all]
$all set beta 0
set seltext "segid WT1 to WT99 and same residue as abs(z) < 25"
set sel [atomselect top $seltext]
$sel set beta 1
set badwater [atomselect top "name OH2 and beta > 0"]
set seglist [$badwater get segid]
set reslist [$badwater get resid]
```

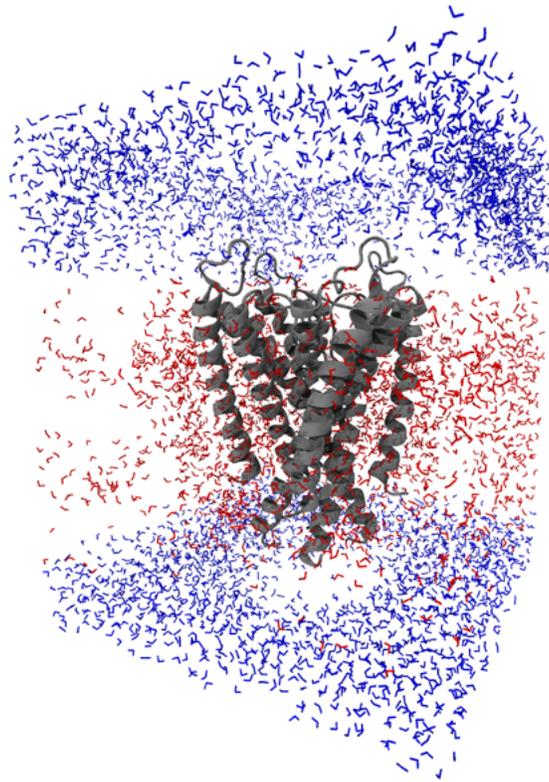


Figure 9: Outcome of solvation. Red water molecules are located in the membrane region and must be deleted.

```

mol delete all
package require psfgen
resetpsf
readpsf kcsa_popc_water_TEMP.psf
coordpdb kcsa_popc_water_TEMP.pdb
foreach segid $seglst resid $reslist {
  delatom $segid $resid
}
writepdb kcsa_popcw.pdb
writepsf kcsa_popcw.psf
file delete kcsa_popc_water_TEMP.psf
file delete kcsa_popc_water_TEMP.pdb

```

- 6 Load files `kcsa_popcw.psf` and `kcsa_popcw.pdb`. Check that your system is OK and, if necessary, regenerate the files by using the script

```
solv_remove.tcl.
```

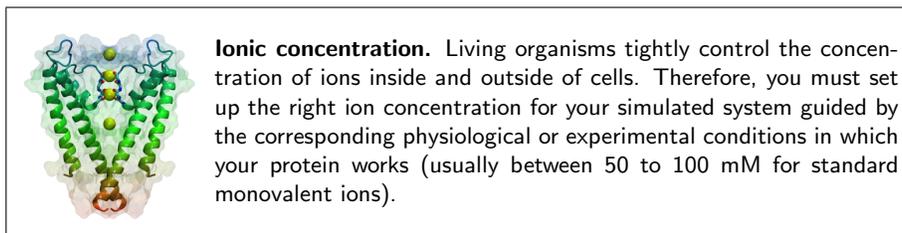
Now we will use the *Autoionize* Plugin, which creates a specified ionic concentration of either KCl or NaCl by transmuted random water molecules into ions. The plugin also neutralizes the system, which is important for NAMD's long-range electrostatics to work properly. Note that the ionization of your system must be done when everything else (protein, membrane, water) is ready.

- 7 Select Extensions → Modeling → Add Ions in the VMD Main window, then enter the following parameters:

Option	Setting
Input PSF	kcsa_popcw.psf
Input PDB	kcsa_popcw.pdb
Output prefix	kcsa_popcwi
Concentration	0.4
Neutralize	Yes
Min. distance from molecule	5
Min. distance between ions	5
Segment ID	ION
Switch to KCl	Yes

Then click Autoionize and wait a minute or two. It's that simple! Note that we have used 0.2 M KCl as the ionic concentration of our system (0.2×2).

- 8 Load the output files of autoionize (*kcsa_popcwi.psf* and *kcsa_popcwi.pdb*) into VMD and check that the ions are really there and that the system is neutral (Fig. 10).



Finally, after all of this work, we are ready for minimization and equilibration with NAMD. This is described in the next unit.

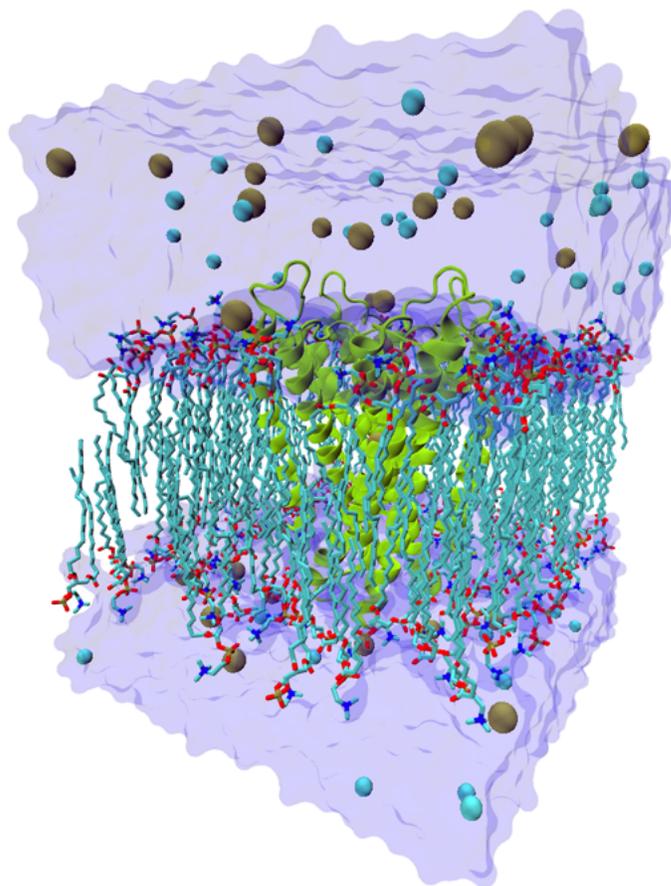


Figure 10: Final simulation system including the KcsA channel, a POPC membrane, water, and ions.

3 Running a Simulation of KcsA

In unit 3, you will use the model containing KcsA embedded in a membrane to perform molecular dynamics simulations and obtain an equilibrated system.

3.1 Melting of Lipid Tails

When using a membrane patch that has not been equilibrated, like those provided by the membrane plugin of VMD, we strongly recommend to perform first a simulation in which everything (water, ions, protein, lipid headgroups) except lipid tails, is fixed. In this way we will induce the appropriate disorder of a fluid-like bilayer. Here we will prepare and run such simulation.

- 1 Set your current directory to `mem-tutorial-files/03-MINEQ/`
- 2 Open the file `kcsa_popcwimineq-01.conf` in a text editor.

We assume you are familiar with NAMD configuration files (otherwise have a look at the NAMD tutorial). However, we will discuss specific points that are important for membrane protein simulations and for this particular simulation of KcsA.

- 3 Locate the section labeled `#Input`. It should look like:

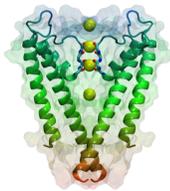
```
# Input
paraTypeCharmm      on
parameters          ../par_all27_prot_lipidNBFIX.prm
```

Note that we are not using the standard CHARMM parameter file. We are using a parameter file that contains the so called “NBFIX” correction for the interaction between carbonyl oxygens and potassium ions, particularly important for the description of ions in the KcsA selectivity filter.

- 4 Open the file `par.all27_prot_lipidNBFIX.prm` in a text editor.
- 5 At the very end of the file, you should be able to find a section labeled NBFIX:

```
NBFIX
!-----Modified potassium ! Free E relative      Interaction E
                          ! to standard K+      with isolated NMA
  0 POT      -0.1021763    3.600750 ! S.Noskov&B.Roux, 2004
  0 SOD      -0.0750200    3.27450  ! -16.25
```

This section overrides the standard Lennard-Jones interactions for oxygen and potassium ions, and for oxygen and sodium ions.



NBFI. The standard parameters for ions have been optimized to reproduce bulk properties of ionic solutions, while the NBFI parameters were optimized for the specific case of ions interacting with carbonyl oxygen atoms as determined by Sergei Noskov, Simon Bernèche and Benoît Roux, *Nature* **431**, 830-834 2004.

6 Close the text editor with the parameter file and go back to the editor with the NAMD configuration file.

7 Locate the section labeled # Periodic Boundary Conditions. It should look like:

```
# Periodic Boundary Conditions
# NOTE: Do not set the periodic cell basis if you have also
# specified an .xsc restart file!
if {1} {
cellBasisVector1    77.    0.    0.
cellBasisVector2    0.    77.    0.
cellBasisVector3    0.    0.    90.
cellOrigin          0.285711854696 0.299352765083 6.22171497345
}
wrapWater           on
wrapAll             on
```

We will check with VMD that the size and center of the system are correct.

8 Open VMD and load the files `../02-MEMBRANE/kcsa_popcwi.psf` and `../02-MEMBRANE/kcsa_popcwi.pdb`. Make sure you load the PDB file into the PSF file.

9 Open the Tk Console window and type:

```
set all [atomselect top all]
measure center $all
measure minmax $all
```

The position of the center of the cell should roughly match the position found in the NAMD configuration file. However, the `measure minmax` command will return values equivalent to cell dimensions that are larger than what we actually have in the configuration file (`{84, 81, 90}` instead of `{77, 77, 90}`). This apparent inconsistency is caused by fragments of lipid molecules, which usually stick out from the boundary of the simulation cell, but are not wrapped unless the center of mass of the whole lipid crosses the periodic boundary. One can avoid this problem by using a selection that does not contain lipids.

10 Type in the Tk Console:

```
set wat [atomselect top water]
measure minmax $wat
```

The output of `measure minmax` should now be consistent with the simulation cell size specified in the NAMD configuration file. It is always desirable to give a bit of extra ($\sim 1 \text{ \AA}$) space between the boundary atoms of your system and the boundaries of the simulation cell.

11 Lets visually check our periodic cell dimensions. Type in the Tk Console:

```
molinfo top set a 77
molinfo top set b 77
molinfo top set c 90
```

12 Select the Graphics \rightarrow Representations... menu item and then click on the Periodic tab of the Graphical Representations window. Select different periodic images to draw and check that the gap between periodic images is small or negligible (use also the cell dimensions obtained when using the “all” selection and check the undesirable large gaps present between periodic images in such a case, Fig. 11).

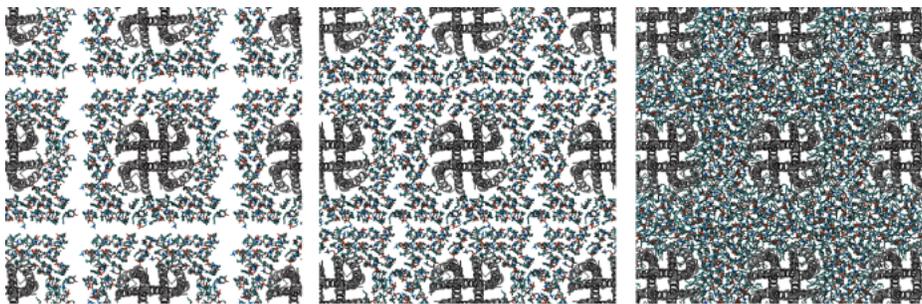


Figure 11: Setting up the right cell dimensions for periodic boundary conditions. Left panel shows periodic arrangement with gaps (overestimated size of the system). Center panel shows the same system but using a better estimate of the simulation cell size. Right panel, periodic arrangement after several steps of minimization and equilibration.

13 Close VMD and return to the text editor containing the NAMD configuration file. Locate the section labeled `#PME (for full-system periodic electrostatics)`. It should look like:

```
#PME (for full-system periodic electrostatics)
if {1} {
PME                yes
```

```
PMEGridSizeX      80
PMEGridSizeY      80
PMEGridSizeZ      90
}
```

Note how we carefully chose the Particle Mesh Ewald grid size so as to have at least one grid point per Å in each direction, and also in such a way that the number of grid points is a multiple of two, three, or five.

14 Now find the section labeled `# Fixed Atoms Constraint`. It should look like:

```
# Fixed Atoms Constraint (set PDB beta-column to 1)
if {1} {
fixedAtoms          on
fixedAtomsFile      kcsa_popcwi.fix
fixedAtomsCol       B
fixedAtomsForces    on
}
```

This section uses the file `kcsa_popcwi.fix` to determine which atoms of the simulated system will be held fixed in space. We will create this file shortly.

15 Go to the very end of the configuration file. You should find the following commands:

```
# Minimization
if {1} {
minimize            1000
reinitvels          $temperature
}
run 250000 ;# 0.5 ns
```

which indicate that NAMD will run 1000 steps of minimization, then it will reinitiate velocities according to the desired temperature, and then it will run dynamics for 0.5 ns (using a 2 fs timestep).

16 Close the text editor.

Before performing the simulation, we need to generate a file indicating which atoms will be fixed.

17 Open VMD and type in the Tk Console:

```

mol new ../02-MEMBRANE/kcsa_popcwi.psf
mol addfile ../02-MEMBRANE/kcsa_popcwi.pdb
set all [atomselect top "all"]
$all set beta 0
set fixed [atomselect top "water or name CLA POT or protein or \
(chain L and name O2 P1 O3 O4 O1 C15 H52 H51 H11 C11 H12 \
N C14 H42 H43 H41 C12 H22 H23 H21 C13 H33 H31 H32)"]
$fixed set beta 1
$all writepdb kcsa_popcwi.fix
exit

```

The commands you typed above created a file, `kcsa_popcwi.fix`, that contains zeros in the Beta field of all lipid tails atoms, and ones in the Beta field of all other atoms.

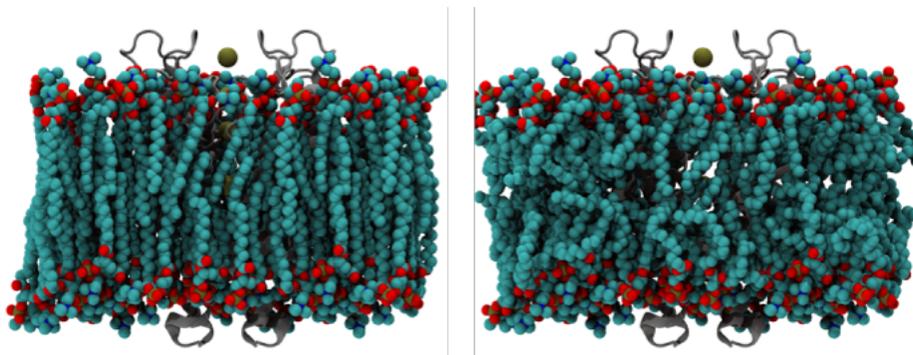


Figure 12: Melting lipid tails

- 18** Run your simulation using NAMD2. Usual NAMD installations will accept the command line
- ```
namd2 kcsa_popcwimineq-01.conf > kcsa_popcwimineq-01.log
```
- in a terminal window. Check your local installation of NAMD and run it accordingly.



**Running.** The solvated system containing KcsA embedded in a membrane comprises over 40,000 atoms. Unless you have access to a parallel computer with 10 or more nodes, we suggest that you look at the trajectories provided in the example-output directory.



**Warning.** The configuration files provided in this tutorial are designed for users with limited computational resources. Thus, simulations are performed using a 2 fs timestep, the SHAKE algorithm for hydrogen atoms, and a multiple time stepping algorithm for the computation of long-range electrostatic interactions. Ideally, one should use a uniform 1 fs time step without SHAKE. This is particularly important for long simulations performed at constant energy. However, you may want to still keep water molecules rigid, as the CHARMM force field was parameterized using rigid water molecules.

- 19 Once your simulation is done, open VMD and load the trajectory `kcsa_popcwimineq-01.dcd` on top of the corresponding PSF file `kcsa_popcwi.psf`.

If your simulation was successful, lipid tails should now look disordered (Fig. 12).

### 3.2 Minimization and Equilibration with Protein Constrained

We are not quite ready for running the system with full dynamics. Because we have put our system together by hand, it has many unnatural atomic positions. If we tried to do a dynamics run now, it would immediately fail, with many atoms moving at very high velocity. Thus, our second run with NAMD will be a “minimization” run, which simply guides the system to the nearest local energy minimum in configuration space. Minimization will be then followed by an equilibration with the protein constrained, so as to permit the environment to relax first.

- 1 Set your current directory to `mem-tutorial-files/03-MINEQ/`
- 2 Open the file `kcsa_popcwimineq-02.conf` in a text editor.
- 3 Check that the `outputName` has been changed appropriately.
- 4 Go to the section `# Continuing a job from the restart files:`

```
Continuing a job from the restart files
if {1} {
set inputname kcsa_popcwimineq-01
binCoordinates $inputname.restart.coor
binVelocities $inputname.restart.vel
extendedSystem $inputname.restart.xsc
}
firsttimestep 251000
```

Note how we have activated this section (“`if{1} {`” instead of “`if{0} {`”) so NAMD can read the restart files from the previous run. The counter of time steps has also been updated.

- 5 Check that the line “`temperature $temperature`” has been commented out (with a `#`).

Now that velocities (which determine temperature) have been read from the previous run, we do not need to set the initial temperature here. Note also that information about the periodic cell is also read from a restart file, and therefore the `# Periodic Boundary Conditions` section has been deactivated.

- 6 Because we will no longer fix atoms in the system, check that the `# Fixed Atoms Constraint` section has been deactivated as well.
- 7 Locate the section labeled `## EXTRA PARAMETERS`. There are two additional sets of parameters in this section:

```
constraints on
consexp 2
consref kcsa_popcwi.pdb
conskfile kcsa_popcwin.cnst
conskcol B
margin 3

tclforces on
set waterCheckFreq 100
set lipidCheckFreq 100
set allatompdb ../02-MEMBRANE/kcsa_popcwi.pdb
tclForcesScript keep_water_out.tcl
```

The first set of additional parameters impose harmonic constraints on the protein. Harmonics constraints are applied using a PDB file that tags the atoms to be restrained. This file will be generated below. The use of such harmonic constraints permits lipids, water, and ions to adapt to the protein in its crystal form.

In addition, we use the `margin 3` command to allow large volume fluctuations that are typical during the first dynamics of a new system in an NpT ensemble. Otherwise, the simulation will likely finish with the message: **FATAL ERROR: Periodic cell has become too small for original patch grid!** (in which case you should just restart your simulation from the last saved restart files).

The second set of additional parameters calls a Tcl script, named `keep_water_out.tcl`, which prevents hydration of the membrane-protein interface during equilibration.



**Pumping out water.** The `keep_water_out.tcl` Tcl script automatically applies forces to water molecules entering the membrane hydrophobic region. The script makes use of the NAMD Tcl Forces module. We encourage you to look at the script, use it, and modify it to suit your needs. A detailed description of the script can be found in the “User-Defined Forces in NAMD” tutorial. Note how we modify it to exclude water molecules present in KcsA cavities.

Before performing the simulation, we need to generate a file for the harmonic constraints on the protein.

8 Open VMD and type in the Tk Console:

```
mol new ../02-MEMBRANE/kcsa_popcwi.psf
mol addfile ../02-MEMBRANE/kcsa_popcwi.pdb
set all [atomselect top "all"]
$all set beta 0
set prot [atomselect top "protein"]
$prot set beta 1
$all writepdb kcsa_popcwi.cnst
exit
```

Here we used the coordinates stored in `kcsa_popcwi.pdb` because in the previous run all protein atoms were fixed. Otherwise, we would have used the restart file `kcsa_popcwimineq-01.restart.coor` instead.

The commands you typed above created a file, `kcsa_popcwi.cnst`, that contains zeros in the Beta field of all atoms except those that belong to the protein, which contain a 1. The latter value corresponds to the spring constant “k” of the applied harmonic constraint in kcal/mol/Å<sup>2</sup>. Note that you can easily estimate how much atoms will move around their constrained position by using the equipartition theorem and a given spring constant:

$$\frac{kx^2}{2} \sim \frac{k_B T}{2} \Rightarrow \Delta x \sim \sqrt{\frac{k_B T}{k}} \quad (1)$$

9 Run your simulation using NAMD2. Usual NAMD installations will accept the command line

```
namd2 kcsa_popcwimineq-02.conf > kcsa_popcwimineq-02.log
```

in a terminal window.

### 3.3 Equilibration with Protein Released

After minimization and equilibration with the protein constrained, we hopefully have a system in which lipids are well packed around the protein, while water has not entered forbidden regions (Fig. 11 and Fig. 13). We will now proceed

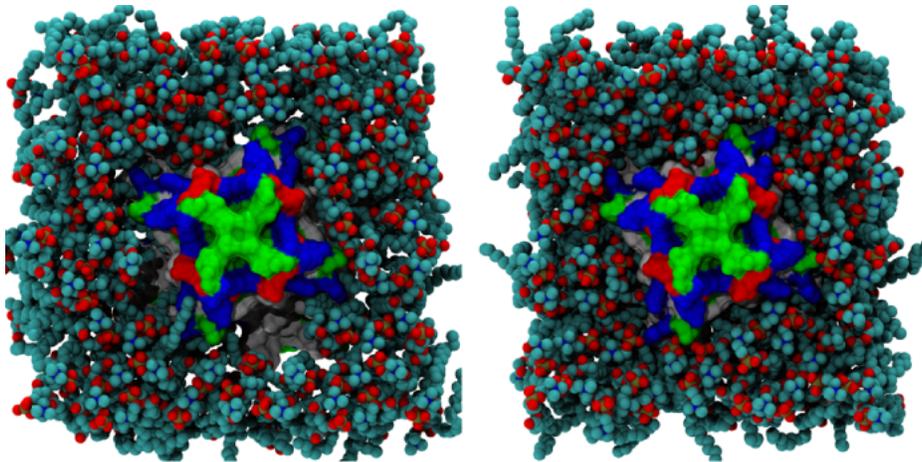


Figure 13: Achieving a good packing of lipids around the protein. Note the gaps in the membrane–protein interface (left panel) before proper equilibration (right panel).

to release the harmonic constraints and further equilibrate the whole system.

- 1 Open the file `kcsa_popcwieq-03.conf` in a text editor.
- 2 Locate the section labeled `# Continuing a job from the restart files` and note how we use the output files of our previous run to set starting coordinates, velocities, and simulation cell size.
- 3 Go to the `## EXTRA PARAMETERS` section and check that the restraints are actually not there. The Tcl script `keep_water_out.tcl` is also not called this time. The command `margin` is eliminated as well, because large volume fluctuations are not expected after the initial equilibration.
- 4 Look at the end of the configuration file. We have eliminated the minimization step in the present simulation.
- 5 Run your simulation as indicated above, using now the NAMD configuration file `kcsa_popcwieq-03.conf`.

After the simulation is done, your system should be equilibrated fairly well. You can monitor the stability of the protein through the computation of RMSDs and by looking at the resulting trajectory with VMD.

### 3.4 Production Runs

Now that the protein has been equilibrated, we are ready to perform production runs. There will be one main difference with the previous simulations you have already performed.

- 1 Open the file `kcsa_popcwieq-04.conf` in a text editor.
- 2 Locate the section `# Constant Pressure Control`. Note that we have set the `useConstantArea` command to `yes`.

Simulation of membrane bilayers have shown that the current CHARMM force-field parameters do not reproduce the experimentally observed area per lipid over long MD trajectories. During the previous simulation steps, we let the area in the xy-plane fluctuate to permit packing of lipids against the protein. However, after good packing has been observed, one should keep the area in the xy-plane constant. Ideally, one should also compute the effective area per lipid in the system by computing the area of the simulation cell and subtracting from it the protein area.

- 3 Run your simulation as indicated above.
- 4 Look at the trajectory with VMD by loading PSF and corresponding dcd files.

Production runs may feature additional parameters or employ different computational techniques that suit the purpose of the investigation being carried out. The outcome of the simple production run presented here can already be analyzed in a number of different ways. For instance, you may want to try different VMD tools that allow you to compute density of water molecules, ions, or lipids, as well as electrostatic potentials, RMSD, motion of center of mass and diffusion of molecules.



**Final Homework.** Set up a simulation of the Sodium Fo-ATP synthase in POPC. The pdb code of the protein is 1YCE. Use only one C11 ring (there are four in the pdb database), use DOWSER, and place four lipid molecules inside of the ring (two on each side).

This ends the Membrane Proteins tutorial. We hope that you learned a lot with it, and that you will make a great use of all the capabilities VMD and NAMD have to offer for simulations of membrane protein systems.

## Acknowledgements

Development of this tutorial was supported by the National Institutes of Health (P41-RR005969 - Resource for Macromolecular Modeling and Bioinformatics).