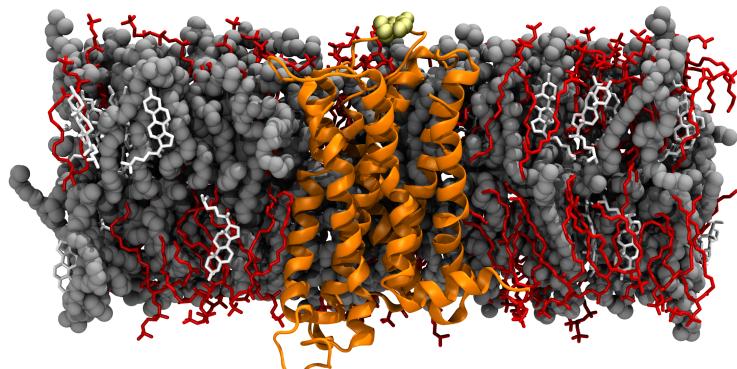


# ProLipids Midsummer Symposium

## Module A: Introduction to Biomolecular Simulations from Theory to System Preparation

Juho Liekkinen      Petri Kaurola      Giray Enkavi      Waldemar Kulig  
Matti Javanainen      Sami Rissanen

June 11, 2019



### General Work Flow of Computational Research

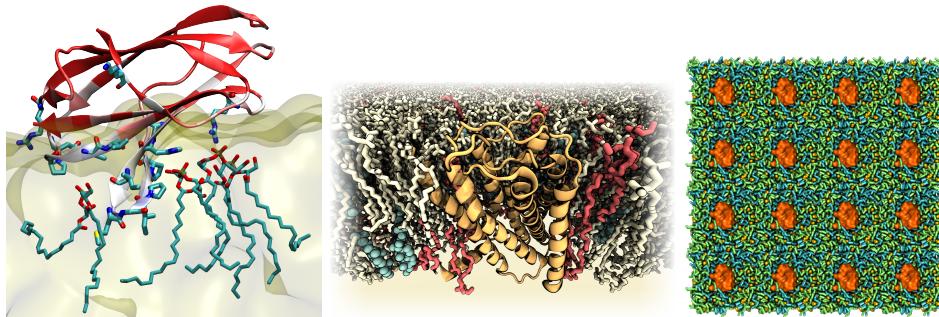
The goal of this module is to provide an idea of what kind of information can be extracted from molecular dynamics simulations, what are the most commonly studied properties of the simulated systems and how they can be compared with experiments. Studying biological systems with molecular dynamics method involves many steps and for completeness we focus here on the preparation, simulation, and the analysis. The general work flow of computational research is approximately as follows:

1. Study the literature, understand the problem
2. Select a suitable model (atomistic, coarse-grained)
3. Parametrize missing molecules
4. **Construct the initial structures of the system**
5. Select simulation conditions
6. Run the simulations on a supercomputer
7. **Analyze the obtained trajectories**
8. Compare with experiments to verify your findings
9. Write a paper and publish it

## Contents of this Module

This module is divided into three labs that study various phenomena involving lipid–protein interactions. These are listed in below:

1. Absorption of Peripheral Proteins to Membrane
2. Constructing the Simulation system
3. Intramembrane Protein–Lipid Interactions and Their Effect on Protein Conformation



The estimated schedule for each lab (a total of 2 hours and 15 minutes) is:

- 10 minutes of describing the scientific background and research questions with a casual discussion and brainstorming on the feasibility to tackle the questions using experimental techniques
- 115 minutes of hands-on exercises including visualization and analysis guided by instructions and instructors, with the aim to answer the posed research questions
- 10 minutes of wrap up and discussion of the results with conclusions

The learning outcomes of this module are:

- Understand the possibilities and limitations of classical MD simulations
- Perform simple analysis and visualization of molecular dynamics trajectories
- Construct simulation systems
- Link experimental findings and simulation results

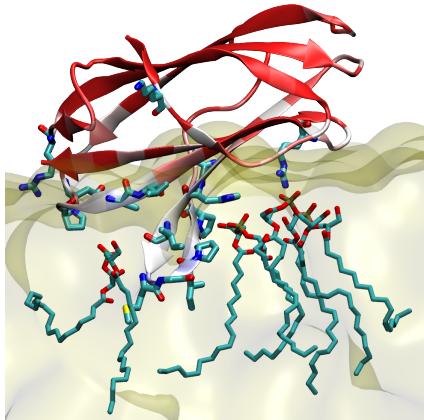
In this document, the important parts are highlighted as:

Notes

Questions

Bonus tasks

## Lab 1: Adsorption of Peripheral Proteins to Membrane



### 1.1 Scientific Background

Peripheral membrane proteins are amphipathic proteins, which adsorb onto the membrane surface reversibly. They are functionally diverse, working as carriers of hydrophobic molecules such as lipids, as electron carriers in the electron transport chain, and as hormones. The mechanistically important aspects of their function involve

1. specific protein–lipid interactions that stabilize binding,
2. the protein conformational changes upon membrane binding,
3. the local perturbation on the membrane structure,
4. and in the case of carriers, how the above listed are coupled to uptake and release of the ligand.

Most, if not all, of the above mentioned mechanisms of various proteins have been studied in atomic scale using molecular dynamics simulations.

In this hands on exercise, we will focus on a peripheral membrane protein, which we have investigated in our lab. Niemann-Pick C2 protein is a cholesterol carrier found in the late endosomes and lysosomes. It is responsible for shuttling cholesterol between the intra-organellar membranes. Its role in cholesterol and lipid metabolism is so important that mutations are not only very rare, but result in a genetic disorder called Niemann-Pick C disease. This progressive neurological condition results in gradual accumulation of lipids in the nervous system and death at a young age.

The aim of this module is to familiarize attendants with the molecular visualization and analysis of molecular dynamics trajectories.

### 1.2 Visualizing the crystal structures

NPC2 was crystallized in *apo* and cholesterol-bound states. We will first examine these structures using VMD. The main aim of this stage is to familiarize ourselves with the structural information and to generate hypothesis driving our simulation design. When you are going through the tutorial, try to focus on the following questions:

#### Questions:

1. What are the structural effects of cholesterol binding to NPC2?

2. Based on the structures, can you come up with any hypotheses on membrane binding mode or residues of NPC2?
3. Can you hypothesize a mechanism for cholesterol uptake and release?

Before we start, open the terminal and go to the directory that contains all the necessary files for *Lab 1*. First, open a terminal window and

```
cd ~/Desktop/ProLipids2019/Lab1
```

### 1.2.1 Loading structures

Using a web browser, go to [www.rcsb.org](http://www.rcsb.org) (Protein Data Bank website), and type **1nep** in the **Search** field. This is the *apo* structure of NPC2. In the top right corner, click

Download Files > PDB Format

and save the file to your current folder (`~/Desktop/ProLipids2019/Lab1`) with a name **1nep.pdb**. To visualize the loaded structure, we will use VMD (Visual Molecular Dynamics) program that is distributed free of charge and runs on MacOS X, Unix, and Windows. Open VMD in terminal:

```
vmd
```

Once VMD is open, let's open the downloaded file:

VMD Main window > File > New Molecule ... >

Click **Browse ...**, select the downloaded file by double clicking **1nep.pdb** and then press **Load**. Now, do the same for the cholesterol-bound NPC2 by downloading **2hka** from the Protein Data Bank and open it to the same VMD session. Remember to make a new molecule as we did for the *apo* structure (**VMD Main window > File > New Molecule ... >**).

*holo*: refers to the ligand-bound protein

*apo* : refers to the protein that is devoid of bound ligand

**Hint:** Normally you can download structures directly using VMD, but due to some structural changes in the PDB server now only the latest release of VMD (1.9.4) can do this.

Once VMD is open, you can directly download the crystal structures of interest by the following steps:

VMD Main window > File > New Molecule ... >

Type **1nep** in the **FileName** field and click **Load**. This will load the *apo* structure.

Notice that in the **Determine File Type** field, various file types are listed. VMD can open many standard file types used in molecular modeling and simulations, and can generally determine the file type from the extension automatically.

Alternatively, you can start VMD and directly load the **pdb** files in terminal.

```
vmd -m 1nep 2hka
```

### 1.2.2 VMD Main Window

Take a look at the **VMD Main** window, you will see that the two molecules that you just loaded are listed. The first column lists the molecule ID, which is important when using the **Extensions > TkConsole** to make complicated atom selections. This, however, is beyond the scope of this tutorial.

The 4 letters on the left of **Molecule** are the *status flags* that you can change.

- T is for **top**, which identifies one of the molecules as default. Many VMD functions will apply to this molecule unless ID is specified specifically. By default, the last loaded molecule becomes **top**. It makes sense to keep the molecule that you want to work on as **top**. Let's make **1nep top** by double clicking under **T** column. You should now see that the **T** attribute is moved.
- The second column **A** identifies the **active** molecule. These are the molecules that will be affected by some VMD tools. We will keep both molecules active. Make sure that you see **A** next to both molecules. Otherwise, activate both.
- D means **drawn**. You can hide a molecule by making D disappear. Let's hide **2hka** for now by double clicking.
- F is **fixed**. When a molecule is **fixed** it is unaffected by the screen translation, rotations, and scaling. We will come to these operations in a minute, but you should keep in mind that regardless of whether a molecule is **fixed**, the screen operations do not change the actual coordinates.

Under the **Molecule** column, you can see names of the molecules, which are by default taken from **Filename**. We can replace them with more intuitive names. Right click on **1nep**, click **rename**, and type **apo**. Go ahead and also change **2hka** to **chol-b**.

The other fields show number of atoms (Atoms), number of frames (Frames), and if the volumetric data is loaded (Vol).

### 1.2.3 Translating, rotating, and zooming

Remember that we hid the **chol-b** (previous name **2hka**) and made **apo** (previous name **1nep**) **top** (T) in the previous section. So you should be only seeing the **apo** molecule in the **OpenGL Display**.

First, reset the view to center the molecule on screen.

```
VMD Main > Display > Reset View =
```

Let's also switch to **Orthographic** view, so that we can look at the molecule without distortion.

```
VMD Main > Display > Orthographic
```

Since we are working with a small molecule, **Depth Cueing** can be intrusive (shades the particles that are farther).

```
VMD Main > Display
```

and uncheck **Depth Cueing**.

Familiarize yourself with **VMD Main > Display**. Notice that you can change lighting, hide axes, and rendering mode here.

Now, come back to the **OpenGL Display** and left click on an empty spot. We will practice some transformations. See what the following does?

- While holding the left mouse button move the cursor around.

- Roll the mouse wheel.
- Hit **T** on the keyboard. Notice that the cursor changes to a *hand*. Holding the left mouse key, move the cursor.
- Holding the right mouse key, move the cursor (large movement to left/right may be needed).
- Hit **R** on the keyboard. Holding the left mouse key, move the cursor.

#### 1.2.4 Representations

The default representation in VMD is **line**, where bonded atoms are connected by lines and each atom is colored based on its **name**. To change the representation,

VMD Main > Graphics > Representations

- Make sure that **apo** is listed in the **Selected Molecule** drop-down list in the **Graphical Representations** window.
- Replace **all** with **protein** in the **Selected Atoms** field.
- Select some representations from **Drawing Method** drop-down list to see what each representation looks like.
- Finally, select **New Cartoon**, which is one of the most commonly used representations for proteins.
- Now, select **ResType** from the **Coloring Method** drop-down list. This coloring method shows acidic residues in red, basic residues in blue, polar ones in green, and nonpolar ones in white.
- Try coloring the molecule by **Beta** now. This method colors atoms based on their *crystallographic temperature factor* reported in the **pdb** file.
- Default color scale in VMD is **RWB**; that is, the low values are colored red and high values are colored blue. We can change this by

VMD Main > Graphics > Colors

Select **Color Scale** tab. Change the **Method** and see how coloring is affected. You can close the **Colors** window now.

**Bonus task:** In the **Color Controls**, you have many options to change coloring. Try making the background of **OpenGL Display** white.

- Color the protein using **Graphical Representations** by **Secondary Structure**. Can you identify which colors correspond to which secondary structures?
- The crystal structure also contains atoms that do not belong to the protein, such as water oxygen atoms. To show them, create a new representation by clicking **Create Rep** in **Graphical Representations** window and type **water** in the **Selected Atoms** field. Set **Coloring Method** to **Name** and **Drawing Method** to **VDW**.
- There are other atoms that are neither protein nor water in the crystal structure. While these can sometimes be ligands, they can also be remnants from the crystallographic conditions. We will now visualize and identify these atoms. First, hide the water representation by double clicking on it, so that it turns red. Then, create a new representation and type **not protein** and **not water** and press **ENTER**. Change **Drawing Method** to **Licorice**.

- You should be seeing two molecules. Some of you might be able to identify these compounds, but we can learn about these residues from VMD. Hit 1 on your keyboard and you will see that the cursor changes to *target* sign on the OpenGL Display. Now, click on one of the atoms of either residue. You will see a green tag appear with the residue name, number and the atom name. You can also go to VMD Main > Graphics > Labels to get more detail about the atom. So what are these compounds?
- One of them is a phosphate ( $\text{PO}_4$ ), while the other one is a N-linked glycan (NAG). Let's try to identify which protein amino acid is linked to the N-linked glycan. We will make a selection based on distance. Create a new representation and type `protein and within 3 of resname NAG`. This selection text tells VMD to select protein atoms (`protein`) whose distance with NAG (`resname NAG`) is less than or equal to 3 Å. Notice that only some atoms of the amino acid is covered by this selection.
- Sometimes we would rather show the whole amino acid. To do this, change the selection text so that it reads `protein and same residue as within 3 of resname NAG`.
- Now, click on one of the atoms on the appearing residue to identify what it is. Note that this will only work if your cursor is in atom picking mode. If not, press 1 on your keyboard as we did earlier. What are the residue name and number of the amino acid? Remember that VMD uses 3-letter code for amino acids.

**Bonus task:** Try showing the amino acids within 5 Å of phosphate. Remember, phosphate is selected by `resname P04`. Identify these amino acids by their name and ID.

- We will also try to identify whether there are any disulfide bridges in the protein.

Disulfide bridges are formed between two cysteine residues that are close to each other in the tertiary and quaternary structures of the proteins. Cysteine pairs involved in the disulfide bridges are often known experimentally even before the crystal structures are available. We can also guess whether a disulfide bridge is present by examining the crystal structure for closeness of the cysteine residues.

- Now, create a new representation and type `resname CYS`. This will show all cysteines.
- Let's also change the Drawing Method to CPK and set Bond Radius to 1.1.

Notice that VMD already detects the disulfide bridges based on the distance and connects the cysteines that are potentially forming disulfide bridges with a bond.

- Now, go to VMD Main > Graphics > Labels.... Make sure that Atoms are selected in the drop-down menu on the top left corner. Hold left mouse button to select all the listed atoms and click Delete. This is only to tidy up OpenGL Display a little. Note that you can also hide the atom labels if you wish.
- Now zoom in to one of the highlighted disulfide bridges and identify the cysteine residues that are involved in the disulfide bridge. Remember to hit 1 if you are not in atom picking mode (cursor appears as *target*). We can also measure distances between atoms, angles and dihedrals by selecting atoms.
  - Hit 2 on the keyboard and select two bonded sulfur atoms (yellow) of the disulfide bridged cysteines. They will appear separately on the list in Labels window. Go

to the **Labels** window and select **Bonds** from the drop-down list on the top left corner. Report distance between the picked sulfur atoms.

- Hit 3 on the keyboard and select three atoms of your choosing (preferentially, the pair of sulfurs and the neighboring carbon). On the **Labels** window, select **Angles** and report the angle. Note that the order of atom picking is important for angles. The second atom in picking order forms vertex of the angle.
- You should also see the bonds, angles, and dihedrals on **OpenGL Display** drawn in dashed lines with the value written. If you cannot see it, that is likely because the molecular representation covers it. Go ahead and hide the relevant representation of cysteines. Now, you should see them. If you did the bonus task earlier and changed the background color to white, you may switch it back to black to see the labels more clearly.

### 1.2.5 Structural Comparison of the Loaded Molecules

Let's move our focus on the **chol-b** molecule now.

- Since the two molecules are similar, it makes sense to transfer the representations we created for the **apo** molecule to the **chol-b** molecule.

**VMD Main > Extensions > Visualization > Clone Representations**

Select **top** or **apo** – which is still the **top** molecule – for the **From Molecule** and select **All Molecules** or **chol-b** for the **To Molecule**. Click **Clone**.

- First, go ahead and hide the **apo** molecule by double clicking **D** on the **VMD Main** window and make **chol-b top** by double clicking on the **T** column.
- Next, **VMD Main > Display > Reset View** = to recenter. Now you should be looking at the crystal structure of **chol-b**, which is composed of three disconnected protein chains in **NewCartoon** representation and colored by **Secondary Structure**. You will also see several compounds that are in **Licorice** representation. One of these compounds appears to be embedded in the protein. Identify the residue name and number of the compound. Each protein domain has a unique chain label. Do you see that compound in all individual chains? Which chains contain this compound?

**Hint:** Pick any atom on the compound to identify the compound. Pick any atom on the protein chain to which you see the compound bound to identify its chain. Check the **Labels** window. Remember, you can also clean up the old picked atoms by selecting and either hiding or deleting them.

- One of the chains that contain the compound is **A**. To compare the **apo** structure with the **chol-b** chain **A**, we will superimpose the two structures based on the positions of backbone atoms. Note that this procedure moves the molecule to remove the translation and rotation, but does not change the internal coordinates.

**VMD Main > Extensions > Analysis > RMSD Trajectory Tool**

In the top most field, replace the contents with **protein and chain A**, select **Trace**, and click **ALIGN** on the top right. This operation applies the optimal translation and rotation on the **apo** to superimpose the **chain A** of  $C_\alpha$  atoms of the **apo** and **chol-b**.

Note that we took advantage of the fact that the only chain in `apo` is A. Now, also click RMSD. This will show you the *root mean square deviation of atomic positions* (RMSD), which is a metric used to quantify the difference between two structures. Report the RMSD value listed next to `apo`. Note that average (avg), minimum (min), and maximum (max) are the same, only because we compared two structures. When molecular dynamics trajectories are analyzed, RMSD is calculated for each frame with respect to a reference structure which can be a frame in the trajectory (usually the first) or a crystal structure or an average structure over the trajectory. In that case, the average, minimum, and maximum are meaningful. This is how one can compare quantitatively how much structures differ from each other.

**Hint:** You can also do the superimposition based on `Backbone`, or `noh` (non-hydrogen / heavy atoms). If you try `noh` in this example, you will likely receive an error message. This is due to inconsistent number of heavy atoms present in each structure. Missing atoms are quite frequent in crystal structures, but their positions can be guessed based on the geometry, as long as most connected atoms are present. On the other hand, hydrogens are almost always missing from the crystal structures due to resolution limitations. Luckily, hydrogen positions are also easy to guess with one caveat. That is, the protonation states of certain residues, such as histidines, aspartates, glutamates, and even lysines have to be guessed prior to setting up the simulations. We usually make an educated guess based on literature and acid dissociation constant ( $pK_a$ ) estimation. If longer stretches of atoms and residues are missing, more sophisticated modeling approaches are necessary. We will not go into detail of building molecular dynamics systems here, yet we emphasize that these steps are essential in preparing systems for molecular dynamics simulations.

- Note that, due to the limitations of GUI tool for the RMSD, we can only compare the chain A of both proteins. However, it is more relevant to compare the chain A from `apo` to chains B and C from `chol-b`, because these are the chains that are bound to the ligand as you previously identified. To allow this comparison, we provide a `tcl` script. The script superimposes `apo` to each chain of `chol-b` and reports the RMSD for  $C_\alpha$ , backbone, and heavy atoms for each chain. Moreover, it assigns the `User` of chain C of `chol-b` field to the heavy atom RMSD *per residue*, which is the RMSD calculated for each residue after the translation and rotation of the whole protein is removed.

You can read the script in your spare time. To use the script

VMD Main > Extensions > TkConsole

Type `source superimpose.tcl`. Compare the RMSD between chains. The final structures are so that `apo` is superimposed on to chain C of `chol-b`.

- Now that we have structures superimposed on each other, we can visually inspect differences. Let's first do some cleaning. Go to `Labels` and clean out all labels in `Atoms`, `Bonds`, and `Dihedrals`. Go to `Graphical Representations` and hide all the representations for both `apo` and `chol-b`. Now create the following representations.
  - `Graphical Representations`; select `chol-b`; create
    - \* `protein and chain C`
    - `Drawing Method: NewCartoon`
    - `Coloring Method: Trajectory > User > User`

**User** field allows user to assign a value for each atom in the structure as a function of frame. `superimpose.tcl` assigns the RMSD *per residue* to the `user` field. We are coloring the selection based on RMSD *per residue* in this representation.

```
* resname C3S
Drawing Method: Licorice
Coloring Method: Name
```

**Hint:** You see two C3S molecules. If you want to show only the one close to the chain C, you can select atoms

```
resname C3S and within 10 of chain C
```

```
* (protein and chain C) and same residue as within 4 of (resname C3S)
Drawing Method: Licorice
Coloring Method: Molecule
```

- Graphical Representations; select apo; create
  - \* protein  
Drawing Method: NewCartoon  
Coloring Method: Molecule
  - \* protein  
Drawing Method: Licorice  
Coloring Method: Molecule

These selections allow us to compare the backbone and even side chains within vicinity of the ligand. Now, take 5 minutes and carefully investigate to see if there are any major differences. Rotate, zoom in, zoom out, pick atoms, create new representations to do this and describe the differences that you observe. Is there any part of the molecule that is different? The coloring by RMSD *per residue* allows us to see the regions of the protein that have larger difference.

#### Questions:

4. Speculate how cholesterol binding is related to the structural deviation between the two structures?

Now, close the VMD session.

#### 1.2.6 Visualizing a complete system

At this stage, we have learned quite a lot about visualization functions of VMD. We will exploit these skills and some extra to inspect initial structures prepared for Molecular Dynamics simulations. The structures are composed of a lipid membrane and NPC2 placed away from the membrane in bulk solvent. Such systems are commonly prepared by simulating the membrane and protein separately for relaxation first, and then, combining them. These setups allow us to investigate and quantify membrane binding.

We provide you with two pdb files. Go through the following items. Try to make representations and answer questions.

- Open two pdb files from the current directory as separate molecules.

```
vmd -m system_1.pdb system_2.pdb
```

- Hide the `system_2.pdb`: VMD Main and double click on D and make `system_1.pdb` top in the VMD Main window (double click on T).
- Switch to orthographic view: VMD Main > Display > Orthographic
- Uncheck Depth Cueing: VMD Main > Display > Depth Cueing
- Open Graphical Representations:  
VMD Main > Representations > Graphical representations
- Select `system_1.pdb` from the drop-down list in Graphical Representations
- Show the periodic neighbors for the default atom selection `all`: Click Periodic tab and check boxes one at a time for `+X, -X, +Y, -Y, +Z, -Z`. Rotate the system in the OpenGL Display: Hold left mouse button. Zoom in/out: Scroll the mouse wheel.

**Note:** We often perform molecular dynamics simulations using Periodic Boundary Conditions (PBC), which allows us to use small systems without having to deal with boundary effects. There are two main implications of PBC.

- What comes out from one side of the simulation box, appears on the opposite site.
- Atoms interact with each other based on the minimum image convention. That is, the distance between two atoms is the shortest one between any periodic image of the atoms. In other words, atoms on the opposite edges of the box are, indeed, neighbors.

- Hide the default representation: Double click on it.
- Graphical Representations; go to Draw style tab; Create Rep
  - Protein  
Selected Atoms: protein  
Drawing Method: NewCartoon  
Coloring Method: ResType
  - Lipids without hydrogen atoms in licorice representation:  
Selected Atoms: noh lipids  
Drawing Method: Licorice  
Coloring Method: Name

**Hint:** VMD contains aliases to refer to various common biological compounds, such as `protein`, `lipids`, `ions`, `water`. It can also recognize various elements by its name such as `carbon`, `oxygen`, `sulfur`. However, it cannot always recognize every compound. In that case, the user has to specifically refer to the compound. In this system, we have three lipid types `POPC`, `PSM` (sphingomyelin), and `CHL1` (cholesterol). You can see these residue types by scrolling through the `pdb` file and can also see various selection options in the `Selections` tab.

- Cholesterol in VDW representation and color by its residue ID:  
Selected Atoms: noh resname CHL1  
Drawing Method: VDW  
Coloring Method: ResID

**Hint:** ResID is a unique for each individual compound that is within the same chain. It is useful when one wants to refer to a particular compound during analysis.

**Questions:**

5. What is the bound state of the protein? If it is in *holo* state (bound to a ligand), what is the residue ID of the ligand? **Remember**, Hit 1 and click on an atom. You can see the details also in VMD Main > Labels.

- Ions in VDW representation:

Selected Atoms: ions

Drawing Method: VDW

Coloring Method: Name

**Hint:** Ions are usually added to the systems in some predetermined concentration and ion types might depend on the system of interest. However, the number of each ion type is chosen to result in a neutral system. This is important for the correct calculations of electrostatic interactions under periodic boundary conditions!

Determine the ion types in this system: Hit 1 on the keyboard, click on a large and a small ball.

**Bonus task:** Did you know that you could take snapshots and save the current VMD state to work on it later?

Save a visualization state:

VMD Main > File > Save Visualization State...

Filename: state.vmd

Visualization states are useful when you want to continue working on the session later. You can restore a visualization state by vmd -e state.vmd. This will load all the necessary files and create the representations that you have previously saved. You can try it now in a new terminal window.

Take a snapshot of the system. make sure that you are viewing the system from a desirable angle. Remember to hit R to go back to rotation mode if you are in atom picking or translation modes.

Change the background color to white:

VMD Main > Graphics > Colors > Display > Background > 8 White

Hide the axes:

VMD Main > Display > Axes > Off

Render:

VMD Main > File > Render...

Filename: system\_1.tga

Start Rendering

Let's now apply the above steps to system\_2.pdb.

- Hide system\_1.pdb: VMD Main and double click on D
- Show system\_2.pdb: VMD Main and double click on D
- Clone representations from system\_1.pdb:

VMD Main > Extensions > Visualization > Clone Representations

From Molecule: system\_1.pdb; To Molecule: system\_2.pdb

- Go to graphical representations to make some changes.

VMD Main > Graphics > Representations...

Selected Molecule: system\_2.pdb

- Let's show lysobisphosphatidic acid (LBPA)
  - Selected Atoms: noh resname LBPA
  - Drawing Method: VDW
  - Coloring Method: Name

**Hint:** Lysobisphosphatidic acid (LBPA) is an anionic lipid that is unique to the late endosomes and lysosomes. It is very important for the cholesterol carrier function of NPC2.

#### Questions:

6. What is the bound state of the protein in system\_2.pdb?
7. How is structure of LBPA different from POPC? **Hint:** You can google molecule structures.
8. Measure the distance between roughly the closest amino acid of the protein and the head group of a LBPA molecule. **Remember:** Hit 2 and select two atoms. You can change the color of the Labels from

VMD Main > Labels > Bonds

Now, close the VMD session.

### 1.3 Qualitative Evaluation of Membrane Binding

We provide here two molecular dynamics trajectories for you to analyze. The trajectories are generated starting with an initial configuration where the protein is far away from the membrane in a random orientation. Indeed, you visualized some of these initial configurations in the previous section.

We preprocessed these trajectories to remove water molecules, so you will not see the water. The trajectories are 400 ns-long and are saved at every 100 ps, so you should see 4000 frames.

Let's first visualize one of these trajectories.

```
vmd struct_1.pdb traj_1.xtc
```

Now use your skills to:

- Show protein in NewCartoon representation
- Show the lipids without hydrogen in Licorice representation.
- Show ions in VDW representation.
- Measure the distance between a protein atom and a lipid atom.
  - Hit 2 and click.
  - Remember to check

```
VMD Main > Graphics > Labels >  
Bonds [drop-down menu] > Graph [tab] > Graph
```

- Use the tcl script to measure the z-distance between the center of mass of the membrane phosphorus atoms and that of the protein C<sub>α</sub> as a function of time.

```
VMD Main > Extensions > Tk Console
```

```
source zdist.tcl
```

The script will save `zdist_1.agr`. Go to the terminal and plot the file by

```
xmgrace zdist_1.agr
```

Now, close the previous VMD session and open a new one, now loading the second system by

```
vmd struct_2.pdb traj_2.xtc
```

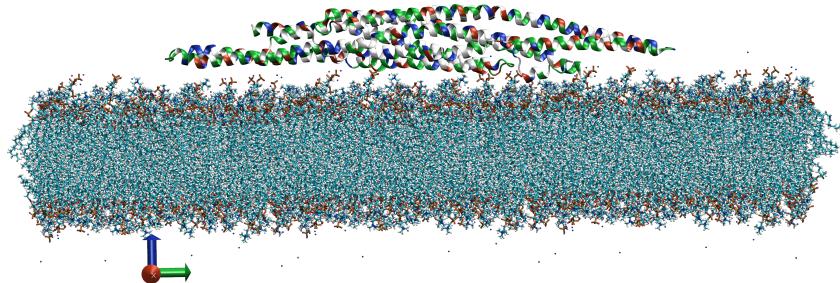
Repeat the above steps and compare the two systems. Do you think these systems are different in terms of membrane binding? You can look at the data you generated using:

```
xmgrace zdist_1.agr zdist_2.agr
```

### Questions:

9. Does the membrane contain LBPA? You can check that by creating a representation with `resname LBPA` and see if any molecules appear?
10. Can you conclude that the protein bind to the membrane in this simulation?
11. If protein binds to the membrane, which residues are involved in binding? You can determine these residues roughly by atom picking (Hit 1 and click).
12. Pay attention to the ions. Do you see anything interesting in their distribution? Are they evenly distributed within the box? Are they interacting with protein or lipids?

## Lab 2: Constructing the Simulation System



### 2.1 Building and Simulating a System With a Protein Solvated in Water

#### 2.1.1 Scientific Background

In the year 2005, Klaus Fütter et al. from the University of Birmingham published a paper in the EMBO Journal, where they had solved a structure for the IRS<sub>p54</sub>/MIM homology domain (IMD) of the insulin receptor tyrosine kinase substrate p53 (IRS<sub>p53</sub>) via X-ray crystallography. This domain has been shown to promote F-actin bundling and, similarly to BAR-domain proteins, it induces and senses membrane curvature. Resembling BAR-domains in structure, IMD is also called I-BAR domain. Therefore, since it is relevant for both actin formation and protein-membrane interactions, it is important to figure out the mechanisms of I-BAR interactions. By using the resolved structure by Klaus Fütter et al., we can build a simulation model for the I-BAR in aqueous solvent, which could be inserted close to a lipid bilayer in order to learn about the interactions between I-BAR and any kind of membrane.

In this lab, we learn how to build a system containing a protein in aqueous solution for MD simulations. We start by downloading and familiarizing ourselves with the protein structure (I-BAR). After slight modification of the resolved structure file, we prepare and take a look into the topology file for the protein. Next, we place the protein in a solution of water and physiological concentration of sodium chloride. Finally, we minimize the energy of the system and make a short MD simulation to obtain a stabilized model for the solvated I-BAR. This can then be further used to build a model where I-BAR interacts with a lipid bilayer.

#### 2.1.2 Downloading, Examining, and Fixing the Structure

First move to the directory for this part of the module. To do this, open a terminal and type

```
cd ~/Desktop/ProLipids2019/Lab2/protein
```

Then, go to the Protein Data Bank and find the protein structure of N-terminal part of the IRS<sub>p53</sub>, also known as I-BAR and given Protein Data Bank identifier 1Y2O. Using a web browser, go to [www.rcsb.org](http://www.rcsb.org), and type 1y2o in the Search field. In the top right corner, click

Download Files > PDB Format

and save the file to your current folder (`~/Desktop/ProLipids2019/Lab2/protein`) with a name `1y2o.pdb`.

**Hint:** In the .pdb file format, there are several different entries describing many features of the structure beyond the scope of this tutorial. You can find more information about the format e.g. from here:

<https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html>

In the same folder, you will find some files which will be required later, as well as a solution folder, which contains all the steps and commands used during this part of the tutorial. Use the solution folder only if necessary.

Now, let us visualize the structure by using VMD. In the terminal, type

```
vmd 1y2o.pdb
```

Change the graphical representations to make it easier to examine the structure

VMD Main window > Graphics > Representations...

We have only one structure open in VMD, so the I-BAR structure is highlighted by default, so our changes in the representation become visible immediately.

Graphical representations window

Hide or delete the representation for all atoms.

- > Create Rep.
- > Write in the Selected Atoms field chain A
- > Coloring method > Select ColorID 1
- > Drawing method > Select NewCartoon
- > Click Create Rep from the top-left corner
- > Write in the Selected Atoms field chain B
- > Coloring method > Select ColorID 4
- > Drawing method > Select NewCartoon

We can see that two I-BAR monomers, both consisting of  $\alpha$ -helices coiling around each other, actually form an anti-parallel dimer (the termini of the monomers face opposite directions). Through this visual inspection, we can say that I-BAR is an elongated anti-parallel all-helical coiled coil homodimer.

Moving on with the VMD, create the following representation which reveals any water in the system:

Graphical representations window

- > Create Rep.
- > Write in the Selected Atoms field water
- > Drawing method > Select VDW

Apparently, the resolved structure contains also idle oxygen atoms, revealed by the water selection. These are of no use to us. There are many ways to remove them, but for the sake of practice, let us do that manually by editing the pdb-file.

For now, close VMD.

You can either use the text editor of your choice, such as vi or emacs (but not word processors like Word) and remove all HETATM entries containing tag HOH (starting from line 4423 until 4603), or you can simply run the following grep command in terminal, which conveniently removes all lines containing HOH strings from a text file and writes the result to a new file.

```
grep -v HOH 1y2o.pdb > 1y2o_water_removed.pdb
```

### Questions:

13. Why do you think there were oxygen atoms scattered around in our protein model?

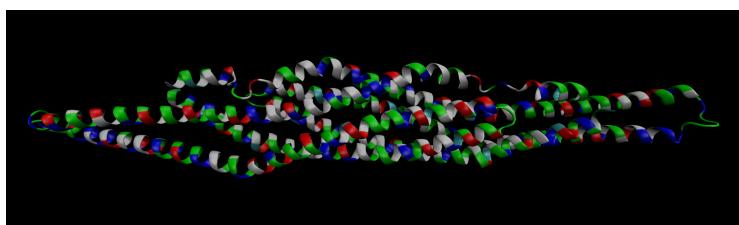
Now, to emphasize the secondary structures and the hydrophobicity of the protein, let's reopen the VMD

```
vmd 1y2o_water_removed.pdb
```

And make the following representation

Graphical representations window

- > Coloring method > Select ResType
- > Drawing method > Select NewCartoon



With this coloring, non-polar residues are white, basic residues blue, acidic residues red, and polar residues green. By examining the representation, you can see that the I-BAR dimer has no obvious consistent hydrophilic or hydrophobic surfaces; you can find polar, charged and non-polar residues all around the protein.

As with many experimental structures, the structure of the I-BAR contains missing or modified atoms and residues. The positions of many missing atoms can be guessed based on geometrical constraints due to neighboring atoms. It is somewhat more complicated to model residues that are completely missing, especially if they are many. This problem can be approached through homology modeling, which is beyond the scope of this module.

The 1y2o structure contains selenomethionines and some missing atoms. For simplicity, we provide you with the ready 1y2o structure 1y2o\_modified.pdb, which we preprocessed (water removal, selenomethionines to methionines, and missing atoms). We will use this processed structure from this point on.

**Hint:** You can find instructions on how to do simple protein modifications with Chimera program by following the link:

<https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html>

### Parametrization of Our Model

Now, we will actually start building the system, so you may close VMD. Throughout this and future computational hands-on assignments, we will use the free command line based molecular dynamics simulation package called GROMACS. Our first task is to prepare the topology of our system. This can be a tedious task, however, luckily GROMACS can do this to standard protein structures automatically. Run the following command and you will be prompted for some input.

```
gmx pdb2gmx -f 1y2o_modified.pdb -o 1y2o_modified.gro -water tip3p -ignh -his
```

Select 15: OPLS-AA/L all-atom force field (2001 amino acid dihedrals) by typing 15 and press **Enter**. Then, the program asks your choice for the histidine protonation states. Select 0 for all 8 prompts.

**What did we do here and what does it mean to prepare topology exactly?**  
First of all, we need to understand where we started; though the `1y2o_modified.pdb` file contained many different kinds of entries, essentially, it only contained the coordinates of atoms belonging to specific residues. This is not enough to describe completely any kind of molecule. We need to know how the atoms are bonded to each other, how strong these bonds are, what are the angles between these bonds, how do non-bonded atoms interact with each other, etc.

All of this information is contained within the topology files (`topol_Protein_chain_A.itp` and `topol_Protein_chain_B.itp`, corresponding to each monomer) we just created. We also prepared a post-processed structure file (`1y2o_modified.gro`), which is analogous to `1y2o_modified.pdb`, but it is simpler and part of the GROMACS distribution format. The third group of files (`posre_Protein_chain_A.itp` and `posre_Protein_chain_B.itp`) contains information with which we can restrain the positions of heavy atoms during simulation if we wish, something we will skip in this tutorial.

After the `pdb2gmx` command was executed, you were asked to choose a force field. It is an important part of the creation of the topology for it describes all the forces between the atoms in our system. There are many different force fields to choose from and it can be a difficult yet important choice that directly effects the result you get from your simulations. When designing a computational simulation, it is imperative to know beforehand, which research questions the simulation is expected to answer. Different force fields perform varyingly in reproducing experimental results, something which a good force field is expected to do well in, yet such force fields may be computationally too demanding. Selecting a force field can be trade-off between performance and applicability, so the choice of a force field has to be justified in order to give proper credit to the produced simulation. In this assignment, we have chosen to use OPLS-AA force fields, which stands for Optimized Parameters for Liquid Systems - All Atom, which is in general known to do relatively well in describing on an atomistic scale biological systems with proteins, lipids, and water solvent.

One more option during our topology creation was the water model we wanted to use. Water is relatively difficult material with unique properties, and many models have been developed over time to reproduce the experimentally solved features of water in simulations. Similarly to the selection of force field, if water interactions are important when it comes to your research question at hand, it is important to choose an appropriately accurate water model. Our choice, `tip3p` is a widely used simple non-polarizable water model with three charge points at positions of oxygen and hydrogen atoms. While you have some freedom in choosing the water model, one should remember that the force fields for biomolecules, such as proteins and lipids, are usually parameterized and optimized against a certain model, which ususally is `tip3p`. For more information on water models, see [en.wikipedia.org/wiki/Water\\_model](https://en.wikipedia.org/wiki/Water_model).

We also gave our command the options for ignoring our hydrogens and choosing histidine protonation states for our system, although this isn't always necessary. GROMACS can add hydrogens to the residues by itself, and by giving the program this option, it can prevent errors during the topology creation. By choosing the histidine protonation states manually, we make sure that both our monomers in our system are identical,

and that the total charge of the system is an integer. We chose neutral charge for all histidines.

### Solvating the Protein in Simulation Box

The first thing we need is to define a simulation box for our system. The following command creates an appropriate size periodic cuboid box (10 nanometers in  $x$  and  $y$ , 25 nanometers in  $z$  direction) for our system, with our protein in the center of it.

```
gmx editconf -f 1y2o_modified.gro -o 1y2o_box.gro -c \
             -box 10 10 25 -bt triclinic
```

Open the `1y2o_box.gro` in VMD to check the simulation box.

```
vmd 1y2o_box.gro
```

Go to VMD Main > Extensions > TK Console. Type in `pbc box` and press Enter to show the simulation box.

After creating the simulation boundaries for our system, we can add water molecules to our system. GROMACS can do this automatically with the following command:

```
gmx solvate -cp 1y2o_box.gro -cs spc216.gro \
             -o 1y2o_solvated.gro -p topol.top
```

The given parameter `spc216.gro` tells GROMACS to use readily equilibrated water structure to fill the box appropriately. The same water structure file could be used when using other three point water models too. We also gave the topology file as a parameter, since GROMACS can automatically overwrite in a new topology file the added water molecules, since each particle of the system has to be listed in the system's topology file. If we were not to give the topology file as a parameter, we could also manually write a new entry in the topology file with the same number of water molecules that `gmx solvate` added.

**Hint:** After running GROMACS commands, the old files are not overwritten, but a new file is created with the same name. The old files retain their names, but `#` symbol is added in front of the names, which indicates that they are the old versions for backup purposes. If you are sure that you don't need the backups, you can safely remove them, for they have no further use in the upcoming steps of building our system.

Next, we will add ions to our system in order to make our simulation more realistic. While doing this, we have to keep in mind a couple of things:

- The concentration of ions should be physiologically relevant
- The simulation system should be neutral for correct treatment of electrostatic interactions. Because our dimer has a net charge of 10 e, we need to add 10 more chloride ions than sodium ions.

You can now run the `grompp` in order to create an input file for `genion`, and then, run `genion`. When running `genion`, you will be prompted to choose the solvent group in your

system, which will contain the particles that will be replaced by ions. Choose `SOL`, short for solvent.

```
gmx grompp -f ionsmdp -c 1y2o_solvated.gro -p topol.top -o ions.tpr
gmx genion -s ions.tpr -o 1y2o_solvated_ions.gro -p topol.top \
-pname NA -nname CL -np 140 -nn 150
```

This command works by randomly replacing some of the water molecules in our system with the ions. With this we have completed building our system. Before finishing with assignment, let us run a brief energy minimization and simulation for our system, just to demonstrate the work flow until the end.

Note that we used 140  $\text{Na}^+$  and 150  $\text{Cl}^-$ . We estimated these numbers from the number of water molecules to achieve the desired concentration of ions (0.1 M). `gmx genion` can also determine this based on the current volume of the system automatically. You can check the option you have by putting `gmx genion -h`.

We also neutralize the system. The 10 extra  $\text{Cl}^-$  were used for this purpose. This is important for correct treatment of electrostatic interactions in the periodic systems.

### 2.1.3 Energy minimization and Dynamics

During our preparation so far, we have been quite forceful with our system. For example, we built missing atoms from residues via external programs, we told GROMACS to add hydrogens as it sees fit, and water-protein interface in our system is only the result of filling chunks of water molecules in our system. No force calculations or dynamics have been involved. Therefore, it is safe to assume, that **our system is very far from equilibrium**. If we were to start simulating the system according to Newtonian dynamics, the forces would be probably be so large that the system might explode.

In order to proceed, before introducing dynamics, we try to minimize the energy in some other way than real dynamics. This can be done e.g. with the steepest descent minimization method, where particles are allowed to drift towards the direction of the negative gradient on the energy landscape. Next, we will run `gmx grompp` command similarly to before, but this time, give as an input not only the solvated system with ions, but also the provided parameter file `minimmdp`, where the steepest descent method has been chosen as the method of energy minimization. Then, we will use the GROMACS `gmx mdrun` engine to actually perform the energy minimization run. The `-v` flag stands for verbose, meaning that you will see the program output on the display, and `-deffnm` flag stands for defining the names for input and output of the run.

```
gmx grompp -f minimmdp -c 1y2o_solvated_ions.gro -p topol.top -o em.tpr
gmx mdrun -v -deffnm em
```

You can see that the program created four files as output. The `em.log` file has the energy minimization process written in it. The new `em.gro` file is the result structure of the energy minimization. The `em.trr` file is the full precision trajectory or the time evolution of the system during the minimization process. The `em.edr` file contains the information of many system observables during the minimization. In order to verify **the success of the minimization**, you can extract the total energy of the system from the `em.edr` file, and look at it visually by plotting. Run the following command

```
gmx energy -f em.edr -o potential.xvg
```

When prompted, select **Potential** and then **0** for ending the prompt.

To plot the output, run

```
xmgrace potential.xvg
```

You should see a drastic decrease in the total energy of the system in the beginning of minimization, and the convergence of the energy towards the end, when the arbitrary criteria for a minimized system are met.

We can now use the energy minimized structure to run a very brief molecular dynamics simulation.

Note that in order to equilibrate our system as well as possible, many successive MD simulations e.g. with different ensembles (first NVT, then NpT) should be run, usually with the solute atoms (in this case the protein) restrained. But for all intents and purposes, let us skip careful equilibration and simply try to run a 1 nanosecond long production simulation, because that will be enough for this tutorial. Also, our system will be further used in a simulation with the membrane, so equilibrating it too much can be meaningless.

The procedure is similar to the energy minimization, but this time, in our `.mdp`, parameters for actual dynamics calculation are defined, e.g. the method for numerical integration for updating the positions and the velocities of the particles (in our case, leapfrog integrator is chosen).

```
gmx grompp -f md.mdp -c em.gro -p topol.top -o md_1ns.tpr
gmx mdrun -v -deffnm md_1ns
```

You can see the remaining time required for the simulation to finish as an output of the `mdrun`. Depending on your computer performance and the size of the system, this can range for this simulation from hours to days. **We will not run this simulation to completion, since the finished structure for this simulation will be provided to you in the future steps required in this assignment. You can cancel the `mdrun` by pressing simultaneously `Ctrl + C`.**

Also, instead of `.trr` file, a less precise but also less space consuming file format for trajectories is produced `.xtc`. The parameters related to this are provided in the `.mdp` file.

This concludes the brief introduction of building and simulating protein in aqueous solvent. Next, you will learn how to prepare membrane structures for simulation.

## 2.2 Building and Equilibrating the Membranes

### 2.2.1 Bilayer Construction

The lipid bilayer we are interested in simulating, consists of two lipid components: DPPC (dipalmitoylphosphatidylcholine) and PIP2 (phosphatidylinositol-4,5-bisphosphate). For this tutorial, we are going to use the atomistic Optimized Potentials for Liquid Simulations (OPLS-AA) forcefield for the lipids. Here, we use only the tools included in the GROMACS 2018 simulation package, some basic Unix text processing commands, and VMD.

First, go to the folder devoted to the bilayer construction

```
cd ~/Desktop/ProLipids2019/Lab2/bilayer
```

By typing `ls` you will see the contents of this folder. All the files required for the construction of our two-component bilayer system are included within the folder. Important files include

- `pip2.gro` and `popc.gro` – Structures of the lipids in human readable and readily visualized format
- `pip2.itp` and `popc.itp` – Include topologies for the lipids
- `topol.top` and `leaflet_1.top` – Topologies of the system needed in the construction steps
- `minimizationmdp`, `compressionmdp`, `eq_nvtmdp`, and `eq_nptmdp` – Molecular dynamics parameters for the steps discussed below
- `remove_waters.sh` and `remove_waters_input.dat` – A script to remove water molecules inside bilayers, with required input parameters
- `bilayer_building_* .sh` – Scripts to process monolayer structures into a bilayer
- `3_equilibration` – Folder containing the files from the equilibration steps

Next, look at the initial structures of the lipid molecules with VMD

```
vmd -m popc.gro pip2.gro
```

This will open both files, and you can toggle which is visible with the "D" (Display) tag in VMD. Toggle between the molecules by hiding the other. Open the **Representations** menu from the VMD Main window:

VMD Main > Graphics > Representations

From the **Selected Molecule** dropdown menu, choose the molecule you want to visualize better. Change the representation for both molecules to CPK and compare the differences and similarities between the lipids. What is the most significant difference? Close VMD.

Next, you will construct the first leaflet of the bilayer. A simple method is to use the GROMACS tool designed for inserting molecules into a simulation system, called `insert-molecules`. First, we will use this tool to place the lipid molecules into a monolayer. In the terminal, type

```
gmx insert-molecules -ci pip2.gro -o pip2_leaflet.gro -box 14 14 1 -nmol 12 \
-rot none -scale 0.3
```

**Note!** Check the GROMACS output displayed in the terminal to make sure you have 12 PIP2 molecules.

The `-ci` option determines what molecules you wish insert into the system, `-box` option determines the size of the box (x y z, in nm) where you want to place the molecules, `-nmol` is

the number of said molecules, **-rot none** makes sure that all our lipids stay in the original orientation, and **-scale** is used to allow the lipids to pack close enough without overlapping. These values can be changed based on the bilayer you want to build.

Next, you will add the POPC molecules in the system. In the terminal, type

```
gmx insert-molecules -f pip2_leaflet.gro -ci popc.gro -o leaflet_1.gro \
                     -box 14 14 1 -nmol 116 -rot none -scale 0.3
```

**Note!** Check the GROMACS output displayed in the terminal to make sure you have added 116 POPC molecules. If you get fewer than 116 POPC, please rerun both of the commands to make sure you get a structure containing a total of 128 lipids, 12 PIP2 and 116 POPC.

Here, the **-ci** option refers to the structure or system, in which you want to add the molecules into. You have now constructed a very simple lipid monolayer containing two components. Visualize this structure with VMD.

```
vmd leaflet_1.gro
```

VMD Main > Graphics > Representations

Choose Drawing Method > VDW.

#### Questions:

14. Is the structure you obtained reasonable?

In order to get more realistic monolayer, we need to perform a short minimization and MD simulation of the system. First, the size of the simulation box (unit cell) needs to be increased in the z-direction to prevent the molecules interacting with themselves. In the command line, type:

```
gmx editconf -f leaflet_1.gro -o leaflet_1.gro -box 14 14 8
```

The box size in the xy-plane remains the same, but in the z-direction the size of the box is increased to 8 nm. Run the energy minimization of this system by first creating a simulation input file with the GROMACS preprocessor (**gmx grompp**). In the command line, type:

```
gmx grompp -f minimizationmdp -c leaflet_1.gro -p leaflet_1.top \
            -o leaflet_1_em.tpr
```

**Note!** GROMACS will give you a note about the non-zero total charge of the system. Only in this and the following step of the construction are you allowed to ignore this message!

And next run the minimization step:

```
gmx mdrun -v -deffnm leaflet_1_em
```

Next, you will compress the monolayer structure to a more compact and physiological lateral density. To this end, run a short compression simulation. In the molecular dynamics parameter (`compressionmdp`) file, note the following:

- `define = -DLIPIDPOSRES` – Use position restraints in the z-direction for PIP2 and POPC. The restraints are included with the lipid topology files (*e.g.* `tail -n 10 pip2.itp`) for the phosphorous atom in the headgroup and the terminal carbon atoms of both sn1 and sn2 tails of the lipids.
- `ref_p = 1000.0 1.0` – Reference pressure in the x-y plane, and z-direction (in bar) in the semiisotropic pressure coupling. Note the high value for pressure in the x-y plane during the short compression step. Longer compression simulation would allow lower pressure
- `compressibility = 4.5e-5 0` – No compression of the simulation box allowed in the z-direction

Assemble the binary simulation input using `gmx grompp`. In the command line, type:

```
gmx grompp -f compressionmdp -c leaflet_1_em.gro -r leaflet_1_em.gro \
-p leaflet_1.top -o leaflet_1_comp.tpr
```

And run the short simulation:

```
gmx mdrun -v -deffnm leaflet_1_comp -dlb yes
```

**Note!** The compression simulation will take several minutes on an average desktop computer. If you wish to save time and move on to the following steps, you can refer to the pre-made structure `leaflet_1_comp_example.gro` and energy file `leaflet_1_comp_example.edr`.

Check the final structure in VMD

```
vmd leaflet_1_comp.gro
```

How does the system look now? Also, analyse the time dependence of the simulation box in x- or y-dimension. To analyze the simulation, run

```
gmx energy -f leaflet_1_comp.edr -o box_size.xvg
```

Choose Box-X or Box-Y by typing in the appropriate number and press `Enter` twice (*e.g.*. Visualize the result with

```
xmgrace box_size.xvg
```

### Questions:

15. Has the system size converged to a specific value?
16. What is the average area per lipid (APL) at the end?

**Hint:** Calculating area per lipid ( $\text{\AA}^2$ )  $\text{APL} = \frac{\text{box}_x \times \text{box}_y}{\text{Number of Lipids}}$

Next, Save the size of the x-y-component of the simulation box to a variable `box_xy`.

```
box_xy=$(tail -n 1 leaflet_1_comp.gro | awk '{print $1;}')
```

Make sure the value of the size of the simulation box is saved to the variable `box_xy` by typing `echo $box_xy`. **You might want to take a note of this number to make sure that it is not modified by mistake.** Next, you need the second leaflet of the bilayer. Use the `gmx editconf` tool to rotate the leaflet from the previous step. At the same time, adjust the simulation box z-component so that in the final structure there is no overlapping molecules. In the command line, type:

```
gmx editconf -f leaflet_1_comp.gro -o leaflet_2.gro \
    -box $box_xy $box_xy 14 -rotate 180 0 0
```

You now have two leaflets of a bilayer in files `leaflet_1_comp.gro` and `leaflet_2.gro` that you will use to construct the system. Open the first and the second leaflet of the bilayer in a single VMD window:

```
vmd -m leaflet_1_comp.gro leaflet_2.gro
```

Change the representation of both systems to VDW. Check if there are overlapping molecules that would potentially cause failure of further simulations. If you notice significant overlap, change the value of z for the `-box` option in the previous `gmx editconf` command.

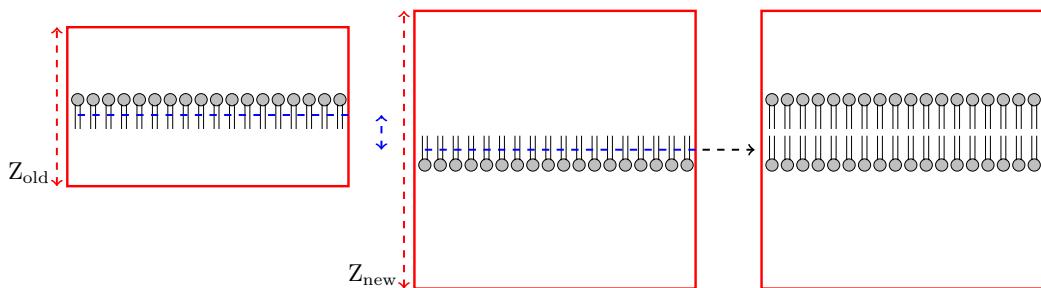


Figure 1: The value for the new value for the z-component is calculated by  $Z_{\text{new}} = (\frac{Z_{\text{old}}}{2} + 3 \text{ nm}) \times 2$ , where  $Z_{\text{old}}$  is from the `leaflet_1_comp.gro`,  $Z_{\text{new}}$  is the size of the new box for the rotated second leaflet (and for the final system), and 3 nm is the estimated difference between the centers of mass of leaflet 1 and 2 (blue lines). In the final system, the coordinates of all of the molecules in leaflet 1 and 2 are copied to `system.gro` with the box vectors from `leaflet_2.gro`, and the number of atoms that is two times the original.

Next, you will construct the actual bilayer. Use the following commands in the terminal. **Type or copy them line by line and press Enter after each.**

```
atoms=$(expr 2 \* $(expr $(wc -l < leaflet_1_comp.gro) - 3))

echo Mixed POPC/PIP2 bilayer > system.gro
echo $atoms >> system.gro
tail -n +3 leaflet_1_comp.gro | tac | tail -n +2 | tac >> system.gro
tail -n +3 leaflet_2.gro | tac | tail -n +2 | tac >> system.gro
tail -n 1 leaflet_2.gro >> system.gro
```

Check the final structure in VMD.

```
vmd system.gro
```

Visually inspect the system and see if any major clashes between molecules occur.

To make sure that there are no problems with the newly constructed bilayer structure, you should run a short minimization.

First, edit the size of the simulation box:

```
gmx editconf -f system.gro -o system_newbox.gro -resnr -1 \
-box $box_xy $box_xy 12
```

**NOTE:** You can check if the `box_xy` is correct by typing `echo $box_xy` in the terminal. Compare this number with what you have written down previously.

The option `-resnr -1` changes the numbering of the residues in the system and removes the duplicate residue numbers. This will make it easier to visualize the system in VMD in the future. Next, use `gmx grompp` to create the simulation input file:

```
gmx grompp -f minimizationmdp -c system_newbox.gro -p topol.top \
-o system_em.tpr
```

And run the minimization:

```
gmx mdrun -v -deffnm system_em -dlb yes
```

You should now have a minimized bilayer structure without water or ions. Next you will add these essential components to your system, after which your bilayer ready for equilibration and production simulations.

### 2.2.2 Adding Water and Ions

Solvating a protein system is a trivial task with `gmx solvate`. Solvating a lipid bilayer or a membrane protein system is not as simple, since the `solvate` program (reminder: you used this tool in the previous section) will fill all the gaps between the lipid acyl chains with water molecules. Especially, the structure you have from the previous steps is not yet well equilibrated and contains a lot of space where water molecules can fit. These extra water molecules will be removed with a custom bash script. Solvate the system as normal:

```
gmx solvate -cp system_em.gro -cs spc216.gro -p topol.top -o bilayer_sol.gro
```

Note the number of water molecules added. The number of water molecules is also updated to the `topol.top`. After solvating, you may visualize the structure in VMD, make atom selection `resname SOL` and you will see many water molecules in the hydrophobic core of the bilayer. To remove the extra water molecules from within the bilayer, run the `remove_waters.sh` script (requires python 3):

```
python3 remove_waters.sh -f remove_waters_input.dat
```

This script prints out how many water molecules it has removed from your system, but does not automatically update the `topol.top`. Using a plain text editor, edit the number of `SOL` in the `topol.top` based on how many waters were removed.

To add the necessary neutralizing counter ions to the solvated system, you will need a `bilayer_ions.tpr` file containing all the parameters for all of the atoms in the system. You can use the previous `minimization.mdp` file for minimization with `gmx grompp` to create the `.tpr` file:

```
gmx grompp -f minimization.mdp -c bilayer_sol_water_removed.gro -p topol.top \  
-o bilayer_ions.tpr
```

Now you have an atomic-level description of your system in a binary file `bilayer_ions.tpr`. Use this file with `gmx genion`, which is the tool to randomly replace solvent molecules with monoatomic ions:

```
gmx genion -s bilayer_ions.tpr -p topol.top -o bilayer_ions.gro -pname NA \  
-neutral
```

Choose group "SOL" for embedding the ions by replacing water molecules.

The previous command should automatically update the `topol.top` and replace the correct number of water molecules with sodium (`-pname NA`) to neutralize the total charge.

**Note!** `gmx genion` may terminate in an error if the `topol.top` file has been edited incorrectly or contains incorrect number of molecules. You may encounter an error if you used the option `-p topol.top` with `gmx solvate` with GROMACS 2018.x. There are a couple of possible measures you can take to avoid the problem. Use only text editors or text editing commands, not word processors like Word, Works, etc. when editing the files. You can open the `topol.top` file in `vi` and remove the `M^`. You can omit the `-top topol.top` from the `gmx genion` command and edit the number of `SOL` and add `NA` according to the printed output.

### Questions:

17. Is the concentration of ions physiologically relevant (~0.2 mol/L)?
18. In case the concentration is not physiological, would you add or remove water molecules to the system to reach a suitable concentration?

#### 2.2.3 Equilibration

**Note!** The next steps are absolutely necessary in the process of building the bilayer. However, unless you have excess time (which you probably don't) skip in this subsection all the actual simulation steps in the equilibrium phase (that is, skip all the `gmx grompp`

and `gmx mdrun` commands (as well as `gmx make_ndx`). Instead, you are given the results from the following steps in the folder

```
~/Desktop/ProLipids2019/Lab2/bilayer/3_equilibration
```

Use these results for the analysis and visualization in VMD.

The bilayer is now solvated and it is electrically neutral. Before any molecular dynamics production runs, we need to make sure that the system contains no steric clashes and the potential energy and internal forces have converged to a minimum. Create the binary input file for the minimization:

```
gmx grompp -f minimizationmdp -c bilayer_ions.gro -p topol.top \
-o bilayer_em.tpr
```

And run the simulation:

```
gmx mdrun -v -deffnm bilayer_em
```

You can analyze the `.edr` file using the GROMACS energy tool:

```
gmx energy -f bilayer_em.edr -o potential.xvg
```

Select "Potential"; zero (0) terminates the input (9 0). Plot this data using XMGRACE. In the terminal type:

```
xmgrace potential.xvg
```

### Questions:

19. Does the potential energy converge to some value?

Now, you should have a reasonable starting structure for equilibration simulations. To begin real dynamics, the lipids, solvent, and ions need to be equilibrated so that the heterogenous system reaches a physiological state where the lipids and the solute are packed correctly. Generally, in the case of lipid bilayers or membrane proteins, a short NVT equilibration phase is followed by a longer NpT phase. In this way, the lipids and the solvent have time to re-orient around themselves and between each other and attain a physiological density and lipid packing. Such process will take some time, and lipid equilibration may take several nanoseconds of simulation time.

First, you will need to create a special index group for the lipids. To do this, use the `gmx make_ndx` tool:

```
gmx make_ndx -f bilayer_em.gro -o bilayer_index.ndx
```

When prompted, type in the following one by one pressing `Enter` after each.

```
2|3
name 13 Lipids
q
```

By entering "2|3" you will merge the groups containing PIP2 and POPC, and with "name 13 Lipids" you will remane this new group as **Lipids**. This new group will be used for the temperature coupling in the following simulations.

Next, create the binary input file for the 100 ps NVT equilibration simulation:

```
gmx grompp -f eq_nvtmdp -c bilayer_em.gro -n bilayer_index.ndx -p topol.top \
-o bilayer_eq_nvt.tpr
```

And run the simulation:

```
gmx mdrun -v -deffnm bilayer_eq_nvt -dlb yes
```

Analyze the temperature progression of the system using **gmx energy**:

```
gmx energy -f bilayer_eq_nvt.edr -o temperature.xvg
```

Choose "**Temperature**"; zero (0) terminates the input. Plot this data using **xmgrace**.

```
xmgrace temperature.xvg
```

The temperature of the system should reach the target value of 310 K quickly, and remain relatively stable around that value.

#### Questions:

20. Could you have used a shorter NVT equilibration simulation to stabilize the temperature?

Now that the temperature is stable, the pressure of the system needs to be equilibrated. The NpT phase for a membrane system is often much longer than equilibration of a protein-in-solvent type of a simulation. Lipid diffusion is relatively much slower compared to water, especially when the initial state of the bilayer is far from the equilibrium. In your bilayer system, there are two main parameters that you need to analyse to confirm that the system has equilibrated: 1) pressure progression, and 2) stable lateral area of the membrane (area per lipid, based on Box-X and Box-Y).

Next, proceed to creating the binary input file for the 10 ns NpT equilibration simulation:

```
gmx grompp -f eq_nptmdp -c bilayer_eq_nvt.gro -r bilayer_eq_nvt.gro \
-t bilayer_eq_nvt.cpt -n bilayer_index.ndx -p topol.top -o bilayer_eq_npt.tpr
```

And run the simulation:

```
gmx mdrun -v -deffnm bilayer_eq_npt -dlb yes
```

This can take quite a lot of time. Therefore, you should kill the process after an couple of minutes by pressing **Ctrl + C**

Analyze the pressure progression of the system using **gmx energy**:

```
gmx energy -f bilayer_eq_npt.edr -o pressure.xvg
```

Choose "**Pressure**"; zero (0) terminates the input. Plot this data using `xmgrace`.

```
xmgrace pressure.xvg
```

The pressure value fluctuates widely over the course of the simulation.

Better indicator of the stability and equilibration of a bilayer system is the lateral area of the membrane. Again, use `gmx energy` to analyze the time progression of either the x- or the y-component of the simulation box, they are the same after all.

```
gmx energy -f bilayer_eq_npt.edr -o box_x.xvg
```

Choose "**Box-X**"; zero (0) terminates the input. Plot this data using XMGRACE.

#### Questions:

21. Has the system size converged to a specific value?
22. How many nanoseconds does it take for the bilayer to reach stable APL?
23. What is the average APL of the bilayer at the end of the equilibration? How does this correlate with the approximate experimental APL values of 67.4 and 68.3 Å<sup>2</sup> for PIP2 and POPC, respectively?

You now have an equilibrated bilayer construction that can be further utilized in more complex simulation systems.

## 2.3 Constructing the Protein–Membrane Systems

We constructed the protein and a membrane patch. Now, we can move on to building the system composed of both the protein and the membrane. Note that the protocol we are following in this module is not set in stone. One can build the protein-membrane system from scratch without separating it into two steps. Or one can build and pre-equilibrate the membrane and, only then, add the protein. Indeed, for a transmembrane protein, one would not want to perform any equilibration of the protein in water. After all, water is not its native environment. There are many options here, and we have chosen this approach to introduce as many tools as possible in building the simulation systems. One should pay attention to a couple of key points regardless of how the systems are built: getting the initial setup as close as possible to the production system and that the simulated molecules remain in physical conformational states.

### 2.3.1 Combining the Protein and Membrane Coordinates

For this part of the tutorial, we'd like to go to the directory named `protein-bilayer`.

```
cd ~/Desktop/ProLipids2019/Lab2/protein-bilayer
```

Here, typing `ls`, you will see `bilayer.gro` and `protein.gro`. These are the equilibrated simulation boxes that you prepared in the previous steps. Our current task is to combine them in a reasonable simulation system.

Let's look at them in VMD, once again

```
vmd -m protein.gro bilayer.gro
```

By now, you should be able to hide water molecules and make selections to show only the lipids and the protein.

Hint: Open the graphical representations window

VMD Main > Graphics > Representations

Create a representation for the `protein` only and change Drawing Method to New Cartoon, color it by Residue type for molecule 0. Switch to molecule 1 from the drop down menu at the top of the Graphical Representations window and create a representation for the resname PIP2 POPC and change the Drawing Method to VDW.

As you can see, the protein is long and the membrane is too small for the protein to interact using its wide surface. Thus, we need a bigger patch.

Now, open another terminal tab by clicking on the terminal window and hitting `Ctrl+Shift+Tab`. You should be in the same directory as you were previously, but if not

```
cd ~/Desktop/ProLipids2019/protein-bilayer
```

The following command will stack three copies of the bilayer system along the *y*-axis and write out a new pdb called `bilayer3.pdb`

```
gmx genconf -f bilayer.gro -o bilayer3.pdb -nbox 1 3 1
```

Go back to VMD and load the new bilayer VMD Main > File > New Molecule > Browse and select `bilayer3.pdb`.

Let's clone the representations that we had for the small patch to the larger patch.

Hint: VMD Main > Extensions > Visualization > Clone Representations. Change From Molecule to 1 and To Molecule to 2 and click Clone.

You can now hide or delete Molecule 1 using the VMD Main window.

We would now like to place the protein as seen in Figure 2. We will do this by rotating and translating the molecule. We do not need to be exact here. The aim of this setup would be to let the system evolve and establish contacts with the membrane. However, we would like to place it in a way that it is close to the final configuration to avoid wasting computer resources.

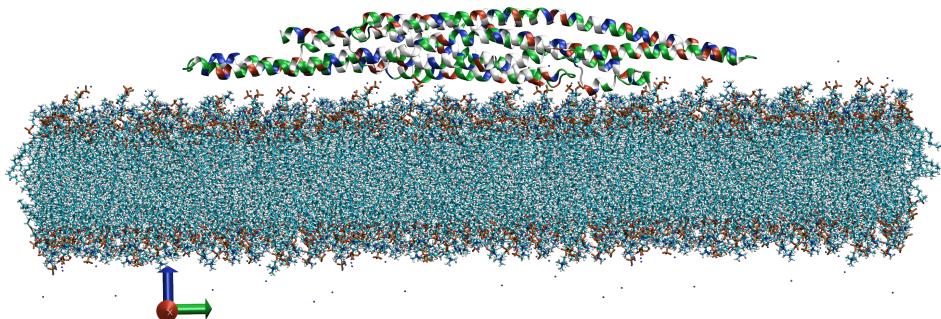


Figure 2: An approximate position and orientation of the protein with respect to the membrane.

Hint: To do this, VMD Main > Mouse > Move > Molecule. Then, **shift + left click** on the protein somewhere around its center and rotate it until you get the flat side of the protein facing the membrane. Then, release the **shift** button to translate the protein until it barely touches the membrane. Now, you can hit the **R** key to go back to screen rotation mode to visualize what you got from all angles.

Once we moved the protein to the desired location, we will save the coordinates of the protein and the bilayer, without the water and ions. Remember that we will have to add the water and ion dynamics later. Water and ion dynamics are relatively fast and re-equilibration happens quickly. However, if there are important buried water and ions in your system, you might want to keep them during this step.

Hint: Go to VMD Main > File > Save coordinates. Make sure that 0: protein.gro is selected from Save data from:. Type protein in the Selected atoms box. Click Save and save it as protein.pdb. Do the same thing for the membrane too, now selecting 2: bilayer3.pdb and typing resname PIP2 POPC in the Selected atoms box. Click Save and save it as membrane.pdb.

We will have to combine the protein and the membrane using some simple one-liners in the command line. Note that this is simply a matter of text manipulation. First close the VMD session. Go to the terminal tab and type

```
cat protein.pdb membrane.pdb | grep ^ATOM > combined.pdb
```

This command simply concatenates the two files and cleans out the headers. One of the important information that a pdb file contains is the box information. Indeed, if you looked at the first lines of a pdb, you will see a line that starts with CRYST1. This is where the information about the box dimensions are kept. Type in the terminal to see for yourself.

```
head bilayer3.pdb
```

The  $x$ ,  $y$ , and  $z$  dimensions of the box is 93.579 280.736 80.722, respectively.

Note that .pdb files use Å for units while .gro files use nm.

For our system, we want to keep the  $x$ ,  $y$  dimensions of the box in agreement with that of the membrane's, but we also want to expand the box in  $z$  to accommodate the protein.

We will do this in two steps. First, we will expand the box in all 3-dimensions.

```
gmx editconf -f combined.pdb -o combined.pdb -bt triclinic -d 1.5
```

Check the dimensions of the new box.

```
head combined.pdb
```

Then, we will reset the  $x$  any  $y$  dimensions of the box to the values for the membrane. Note that the units are in nm unlike the pdb units which are in Å.

```
gmx editconf -f combined.pdb -o combined.pdb -box 9.3579 28.0736 12.1023
```

To check everything worked well, load the molecule in VMD and visualize the periodic images.

Hint: Type in the terminal

```
vmd combined.pdb
```

VMD Main > Graphics > Representations. Go to Periodic and select the periodic images to show.

### 2.3.2 Reintroducing the water and ions

Quit VMD, so that we can solvate the system and generate ions. But before that we will compile a topology based on the membrane and protein topologies we previously generated. The topology simply informs the simulation program GROMACS about the atom definitions and the types of molecules in the system. It should exactly match the order of atoms in the structure (pdb or gro files), which contain the coordinates. We already have this topology compiled for you in the directory. Have a look at it by typing cat topol.top in the terminal.

Now, we can add the water molecules. The following command

```
gmx solvate -cp combined.pdb -p topol.top -o solvated.pdb
```

Now we can perform a minimization. this step will also allow us to check if there are any mistakes in topologies or structures.

```
gmx grompp -f minimizationmdp -c solvated.pdb -p topol.top -o em.tpr
```

Run the minimization

```
gmx mdrun -deffnm em -v
```

We can use same tpr file to also add ions to the system. The following command will add the Na<sup>+</sup> and Cl<sup>-</sup> ions to obtain a concentration of 0.15 M by replacing some random water molecules. It will also neutralize the system and update the topology to include the

amount of ions added. You should type in the appropriate number for the group **SOL** when prompted.

```
gmx genion -s em.tpr -p topol.top -conc 0.15 \
-nname Cl -pname Na -neutral -o ionized.pdb
```

Now we can make another **tpr** file for minimization and perform another minimization for the added ions.

```
gmx grompp -f minimizationmdp -c ionized.pdb \
-p topol.top -o em_ionized.tpr -maxwarn 1
```

Run the minimization.

```
gmx mdrun -deffnm em_ionized -v
```

### 2.3.3 Equilibration simulations in the NVT and NpT ensembles and initializing the production simulations

We can finally run a short NVT and NpT equilibration with the protein restrained to have the solvent equilibrate. And start our production after that. We provide the necessary steps as a bonus task here.

**Bonus task:** Make an index file

```
gmx make_ndx -f em_ionized.gro
```

On the prompt type 21 | 22, which makes a group for the combination of PIP2 and POPC. To rename this group, type **name 26 Lipids**. This index file is necessary for the next step since we define some parameters in the **eq\_nvt.mdp** based on them.

Equilibration in the NVT ensemble with the protein and some lipid atoms restrained.

```
gmx grompp -f eq_nvt.mdp -c em_ionized.gro \
-p topol.top -r em_ionized.gro -n index.ndx -o eq_nvt.tpr
```

Run the simulation.

```
gmx mdrun -deffnm eq_nvt -v
```

Equilibration in the NpT ensemble with the protein and some lipid atoms restrained.

```
gmx grompp -f eq_npt.mdp -c eq_npt.gro \
-p topol.top -r eq_npt.gro -n index.ndx -o eq_npt.tpr
```

Run the simulation.

```
gmx mdrun -deffnm prot -v
```

Production in the NpT ensemble.

```
gmx grompp -f prod.mdp -c eq_npt.gro \
-p topol.top -r eq_npt.gro -n index.ndx -o prod.tpr
```

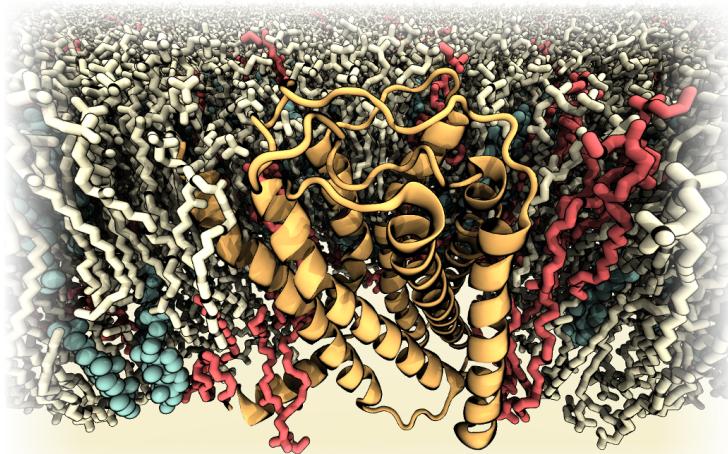
Run the simulation.

```
gmx mdrun -deffnm prod -v
```

**Questions:**

24. How do you think we should check if our equilibration simulations have been sufficient?

## Lab 3: Intramembrane Protein–Lipid Interactions and Their Effect on Protein Conformation



### 3.1 Scientific Background

G protein-coupled receptors (GPCRs) are versatile signaling proteins that mediate diverse cellular responses. With over 800 members, GPCRs constitute the largest family of integral membrane proteins in human genome and represent roughly half of all drug targets in modern medicine.

The human  $\beta_2$ -adrenergic receptor ( $\beta_2$ AR) is one of the best-characterized GPCRs. It is expressed in pulmonary and cardiac myocyte tissues and is a therapeutic target for asthma and heart failure. The functional diversity of  $\beta_2$ AR is associated with its structural dynamics. Recently found structures of  $\beta_2$ AR in the inactive and active states have provided valuable insights into the structure-function relationship of  $\beta_2$ AR. Subsequent biophysical and biochemical studies have provided direct evidences of multiple distinct conformational states for specific GPCRs, such as  $\beta_2$ AR. Intriguingly, it is now evident that the activation of GPCRs is modulated by lipids.

$\beta_2$ -adrenergic receptor is a prototype of cholesterol-interacting GPCRs.  $\beta_2$ AR belongs to the family of class A GPCRs. GPCRs belonging to this class show a high structural similarity and functional diversity. The literature reporting on the specific functional role of cholesterol and other lipids is extensive. It has been experimentally shown that cholesterol affects the conformation and function of many GPCRs. Based on X-ray crystal structures cholesterol has specific contacts with  $\beta_2$ AR, suggesting that  $\beta_2$ AR has binding sites for cholesterol. Spectroscopic and MD simulation studies have reported direct interactions between cholesterol and GPCRs, including  $\beta_2$ AR. Experimental data show that cholesterol binding to  $\beta_2$ AR changes its structural properties. Cholesterol is also necessary in crystallizing  $\beta_2$ AR, and cholesterol and its analogue cholesteryl hemisuccinate (CHS) have been exhibited to improve  $\beta_2$ AR stability. Since the structure and function of GPCRs are closely related, cholesterol binding specifically to  $\beta_2$ AR is also expected to change the functional properties of the receptor. Indeed experimental studies indicate that cholesterol has a functional role in  $\beta_2$ AR. Further, inhibition of  $\beta_2$ AR-associated signaling has been observed with increasing membrane cholesterol content. However, as with GPCRs in general, the atomic-scale mechanism cholesterol uses to regulate  $\beta_2$ AR is not known. Does cholesterol modulate  $\beta_2$ AR activity through membrane-mediated effects by altering the

physical properties of the membrane? Alternatively if regulation takes place through specific direct interactions, then what is the atom-scale mechanism? How do cholesterol analogues (cholesteryl hemisuccinate, oxysterols) interact with  $\beta_2$ AR?

### 3.2 Visualization of Cholesterol Binding Sites in the Crystal Structure of $\beta_2$ AR

$\beta_2$ -adrenergic receptor has recently been crystallized in its active and inactive form. Today we will be working with the inactive form of  $\beta_2$ AR.

First, open a terminal window and go to the directory corresponding to the Lab 2 of this tutorial

```
cd ~/Desktop/Prolipids2019/Lab3
```

Start VMD and open the preloaded structure file from Protein Data Bank (`3d4s.pdb`) of an inactive form of  $\beta_2$ AR. Now, you can do this directly in the terminal window.

```
vmd 3d4s.pdb
```

Visually inspect the protein by

- changing the display perspective

```
VMD Main window > Display > Orthographic
```

- changing graphical representation of the protein

```
VMD Main window > Graphics > Representations ...
```

Change: Selected Atoms to protein; Coloring Method to Secondary Structure; and Drawing Method to NewCartoon.

- creating new representation for the lysozyme Click Create Rep. Change selection to residue 198 to 357; Coloring Method to Color ID and pick color 4 (yellow); Drawing Method to NewCartoon.
- creating new representation for non-protein part of the system Click Create Rep. Change: Selected Atoms to not protein; Coloring Method to Name; Drawing Method to CPK.

#### Questions:

25. How many  $\alpha$ -helices can you see? Does the crystal structure contain all atoms? Are there any ligands bound to the protein? What is the role of lysozyme during the crystallization? Where is the cholesterol binding pocket?

You may close your VMD.

### 3.3 Vizualization of MD Trajectories

In order to answer the questions stated in the introduction we performed a series of atomistic molecular dynamics simulations where the  $\beta_2$ AR was embedded into the lipid bilayers with a different lipid compositions. Here, we consider five different systems (each person will pick one system only) with bilayer compositions listed in the table below:

System	DOPC	CHO	CHS
1	100 mol%	-	-
2	95 mol%	5 mol%	-
3	90 mol%	10 mol%	-
4	60 mol%	40 mol%	-
5	90 mol%	-	10 mol%

where DOPC, CHO, and CHS denote 1,2-dioleoyl-*sn*-glycero-3-phosphocholine, cholesterol and cholesteryl hemisuccinate, respectively.

In order to visualize the trajectories go to the folder corresponding to the system you were assigned to by typing

```
cd systemX
```

where X denotes the system number.

List the content of folder by typing `ls` in your terminal. Files you will find there are:

- `*.gro` – Geometry of the system which can be easily visualize in VMD
- `*.tpr` – Special format file containing all information needed to run the simulation necessary for many analyses
- `*.xtc` – Trajectory file containing the time evolution of the system
- `*.ndx` – Index file containing pre-made selections needed in the analyses
- `topol.top` – Topology of the system.

Take a look at the time evolution of the system using VMD by typing

```
vmd systemX.gro systemX.xtc
```

**Be patient!** VMD loads all frames of the trajectory which may take some time.

Change the display perspective:

```
VMD Main window > Display > Orthographic
```

Now play around a bit with the selections and different representations. Go to

```
VMD Main window > Graphics > Representations ...
```

Replace selection `all` with `resname DOPC`. This will visualize only the DOPC molecules. Similarly, try the selections from the list below.

**Hint:** Remember that you can create multiple new selections and hide existing ones by double-clicking on them.

- `protein`
- `resname DOPC and within 5 of protein`
- `resname SOL`
- `resname DOPC and name "P.*" "N.*"`
- some hints for new atom selections can be obtained from the `VMD Main window > Graphics > Representations > Selections tab`

Remember that you can also modify the looks of each selection in the window below (e.g. **Coloring Methods**, **Drawing Methods**, **Periodic tab**). After playing around for a while, construct your representations such that you can tell apart different lipid molecules and protein.

Once your molecules are colored and displayed nicely, we'll check how they move. In the **VMD Main window**, move the slider (in the bottom) from the right side to the left side and press the **play** button in the lower right corner and you will see that your system start to wiggle around. In the **Graphical Representations** window, go to the **Trajectory** tab and increase the smoothing (separately for multiple groups if needed), and the motion of molecules will be much more eye-pleasing. You may also need to tick **Update selection every frame** and **Update color every frame** in the **Trajectory** tab for each representation you created separately. Now play around with the selections and check out how protein, lipids, ions, and water interact.

#### Questions:

26. Do you see any lipids (cholesterol, cholesteryl hemisuccinate, DOPC) interacting with the protein surface? Where are they localized?

**Hint:** Select lipids of a given type within few Å from the protein surface; tick **Update selection every frame** and **Update color every frame** in the **Trajectory** tab of the **Representations** window.

To analyze protein stability, plot the RMSD of the protein backbone with respect to the initial structure.

- **VMD Main > Extensions > Analysis > RMSD Trajectory Tool**
- tick **Backbone** and click **ALIGN** to superimpose
- tick **Plot** and click **RMSD** to plot the RMSD

#### Questions:

27. Is the structure of the protein stable? Using information from the previous Lab show which part of the protein does move the most?

You may close your VMD.

### 3.4 Density Profiles

After getting some visual clues on the interplay of peptide with the membrane, we move on to quantify the behavior. Every team calculates a few key properties of their system. These are commonly reported in computational papers as they can be compared to experiments and therefore used to validate the model.

First enter the analysis directory of the respective system

```
cd analysis/density_profiles
```

We will calculate the time-averaged mass density profiles of lipid headgroups, protein, lipids (DOPC, CHO, CHS), water, and ions with respect to the *z* coordinate (the membrane normal). In the terminal type

```
gmx density -f ../../systemX.xtc -s ../../systemX.tpr -n density.ndx \
        -ng Y -sl 100 -o density_profiles.xvg
```

**Remember!** Substitute X by the number 1–5 depending on the system you analyze. Substitute Y by the number corresponding to the number of groups you want to analyze. For example if you want to analyze four groups (`protein`, `CHO`, `DOPC`, `SOL`) Y=4, if you want to analyze six groups (`protein`, `CHO`, `DOPC`, `SOL`, `ION`, `lipid_headgroups`) Y=6.

When prompt type the numbers that correspond to the group you want to analyze (after typing each number hit `<Enter>`). `-f`, `-s`, `-n` provide the tool with time evolution of the system, molecular structure and the groups which can be used for selections, respectively. `-ng` sets the number of groups whose density will be calculated and `-sl` selects the number of histogram bins.

Visualize your analysis using `xmgrace` command by typing

```
xmgrace -nxy density_profiles.xvg
```

**Hint:** You might need to change the scale of the Y axis in order to nicely visualize all plots — double-click on the axis and change `Scale` from `Linear` to `Logarithmic` and press `Apply`.

#### Questions:

28. Where is the protein located as compared to the lipid bilayer? Where are the ions? Is water inside the lipid bilayer? Why?

You may close your `xmgrace`.

### 3.5 Volumetric Maps Generation Using VMD

In order to visualize the possible lipid binding sites we will use VMD's VolMap plugin. The VolMap plugin creates volumetric maps (3D grids containing a value at each grid point) based on the atomic coordinates and properties of a specified atom selection.

In the terminal type the following commands

```
cd .../...
```

```
vmd systemX.gro systemX.xtc
```

In VMD Main window > Graphics > Representations ... create representation containing only protein. In VMD Main window > Extensions > Analysis > RMSD Trajectory Tool align the protein using `protein` selection and Backbone Selection Modifier. Finally, in VMD Main window > Extensions > Analysis > VolMap Tool create a selection for chosen lipid type (for example `resname DOPC`), tick `compute for all frames ...` box and click on `Create Map` button. VMD will start calculation of time-averaged volumetric map (it may take some time so be patient!). Play around with the graphical representations (color, size, material, representation, ...) of volumetric map and protein to get nice visualization of lipid-binding sites (if they exist!).

#### Questions:

29. Do you see any lipid-binding sites on the surface of the protein? Where are they? Are they similar to the ones you saw in the crystal structure?

**DO NOT CLOSE VMD** after finishing we will need it in the next step.

### 3.6 Hydrogen Bonds Analysis

In order to calculate the number of hydrogen bonds between the protein and different lipid types we will use VMD's Hydrogen Bonds plugin. If you closed your VMD after finishing previous step of this tutorial, load your trajectory again. If you have your trajectory loaded from previous step, remove all representations in the **VMD Main window > Graphics > Representations** ..., add new representation with **all atoms** selection.

In **VMD Main window > Extensions > Analysis > Hydrogen Bonds** adjust **Selection 1** field to **protein** and **Selection 2** field to chosen lipid type. Pick **all** frames and tick the **Unique hbond**. Click on the **Find hydrogen bonds!** button. New window with the plot showing the number of hydrogen bonds in time will show up after the calculations completes (**Be patient!** might take some time and you don't see any numbers or progressing bar changing before the calculation is finished).

#### Questions:

30. How many hydrogen bonds (on average) are formed between the protein and different lipid types? Is this number stable over time? What does it mean? In the terminal a list of the hydrogen bonds and their occupancy will appear. Which hydrogen bonds are the most stable?
31. Visualize aminoacids and lipid molecules taking part in the bydrgoen bonds using the information given in the terminal. What chemical groups are involved in hydrogen bond formation?

**Hint:** ARG260-Side-NE CHS664-Side-04 25.73% means that relatively stable (occupancy of 25% means that this hydrogen bond lived for about 1/4 of the simulation time) hydrogen bond is created between the nitrogen atom (named NE) of the arginine (**resname ARG**) residue number 260 (**resid 260**) and the oxygen atom (named O4) of CHS molecule (**resname CHS**) number 664 (**resid 664**).

You may close your VMD.

### 3.7 Effect of Cholesterol on Receptor Conformation

Crystallographic studies and previous biophysical and biochemical studies have shown that helices 5–6 (H5–H6) constitute a highly dynamic region of receptor. Upon activation, the most dramatic conformational change, which is conserved among many GPCRs, is a 7–14 Å outward movement of the intracellular end of H6 from the heptahelical core of the receptor. The large rearrangement in the G protein-coupling interface is accompanied by a comparatively subtle change in the ligand-binding pocket. In a conformational change from the inactive to the active state, H5 (around S207) has been found to move inward by 2 Å to establish an optimal interaction between the agonist and the two anchor sites (D113/N312 and S203/S204/S207) on the receptor. Here, we will calculate the distance between the C<sub>α</sub> atoms of D113 and S207 (referred to as L<sub>L</sub>) to measure the displacement of H5 in the ligand-binding site, and the distance between the C<sub>α</sub> atoms of R131 and E268 (referred to as L<sub>G</sub>) to determine the displacement of H6 in the G protein-binding site.

Visualize above mentioned aminoacids inside protein structure. (**HINT:** load your trajectory; create a representation containing protein; create new representation with aminoacids involved in L<sub>G</sub>; create new representation with aminoacids involved in L<sub>L</sub>).

In the terminal type the following command

```
cd analysis/conformation
```

To calculate above mentioned distances we will use the `gmx distance` tool

```
gmx distance -f ../../systemX.xtc -s ../../systemX.tpr -n ca_131_268.ndx \
-oall ca_131_268.xvg -select 'com of group 0 plus com of group 1'
```

```
gmx distance -f ../../systemX.xtc -s ../../systemX.tpr -n ca_207_113.ndx \
-oall ca_207_113.xvg -select 'com of group 0 plus com of group 1'
```

when prompted select groups 0 and 1 (after typing each number hit <Enter>) corresponding to above mentioned residues.

Next, execute the `process.sh` script by typing

```
./process.sh
```

this will process the output files and create the 3D histogram showing the conformational dynamics of  $\beta_2$ AR. Compare your results with the plots created by your neighbors.

**Questions:**

32. Are these plots the same? What is the main difference? What is the conformation of the protein? Do the results depend on the cholesterol concentration? If yes, why is that? Does cholesteryl hemisuccinate have the same effect as cholesterol?

### 3.8 Preparing for the Wrap-Up

Before we wrap up, go through this instruction one more time and make sure that you know the answer to all the questions given at the end of each subsection.