Chapter 5
Lipids

Lipid membranes separate two compartimentes from each other: they separate a cell from the surrounding, or they separate the cytoplasm of cells into organelles. These membranes consist of two layers of lipid, the so called lipid bilayer. The lipid bilayer is a planar, two dimensional fluid.

A large number of proteins belong to the class of membrane proteins. Membrane proteins can be divided into two groups: First, peripheral membrane proteins, which are located on the surface of the lipid bilayer and second, the integral membrane proteins. It is typical for integral membrane proteins, that they are embedded into the phospholipid bilayer. GPCRs belong to the membrane proteins and are also called 7TM receptors, since they consist of 7 transmembrane domains, which cross the lipid bilayer. These transmembrane domains are connected by sections with some few up to some hundreds of amino acids, which are located in the aequeous extra- and intracellular sides of the lipid bilayer.

Within the first molecular modelling studies of GPCRs, the GPCRs were modelled in the gas phase. This was a very rigorous approximation, because, the amino acid side chains, pointing outwards of the receptor, were not in contact with the native surrounding. This could lead to incorrect amino acid side chain conformations, or to artificial interactions between polar or charged amino acids. Additionally, if molecular dynamic simulations were performed of a GPCR in the gas phase, the secondary and tertiary structure of the receptor was not stable. In order to achieve stability, constraints had to be put onto the backbone of the protein. Thus, conformational changes with regard to the whole receptor could not be observed. But with the development of more efficient computers, it was possible to simulate GPCRs in their natural surrounding, like lipid bilayer including intra- and extracellular water. Meanwhile, it is widespread established, to model a GPCR in its natural surrounding.

5.1 Structure of Lipids

Lipids can be divided into several groups, the phosphoglycerides, sterols, sphingolipids, triglycerides and glycolipids. Membrane bilayers are mainly constituted by phosphoglycerides. A schematic representation of phophoglycerides is given in Fig. 5.1.

A. Strasser, H.-J. Wittmann, *Modelling of GPCRs*,
DOI 10.1007/978-94-007-4596-4_5, © Springer Science+Business Media Dordrecht 2013

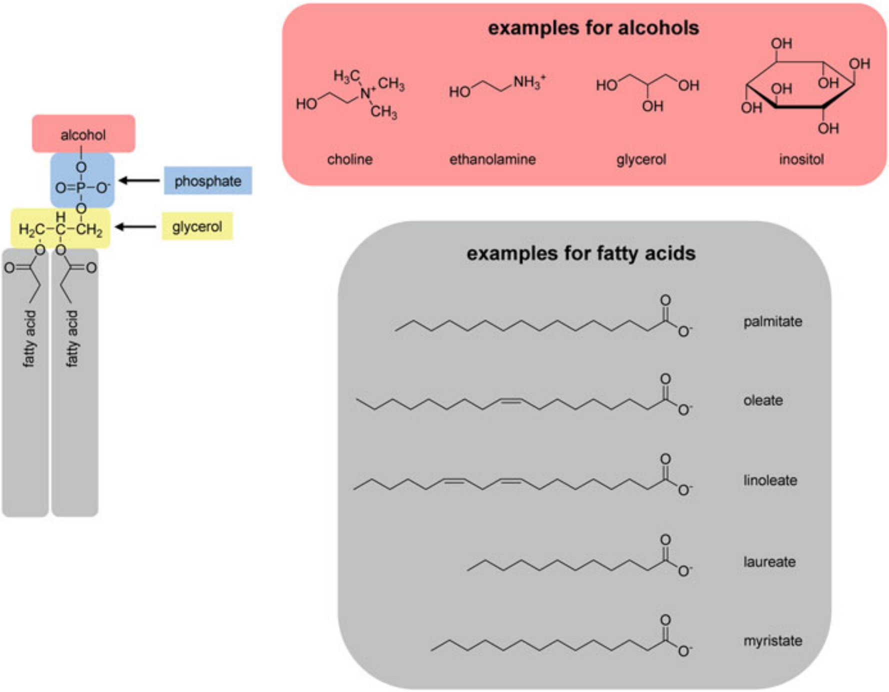


Fig. 5.1 Structure of phosphoglycerides

The phosphoglycerides are established by one glycerol. Two long-chain fatty acids are esterified to the carbons C1 and C2 of the glycerol. The fatty acids are carboxylic acids with about 12–20 carbon atoms. A phosphoric acid is esterified to C3 of the glycerol and an alcohol to the phosphate. Due to their chemical structure, phosphoglycerides are amphiphilic. The head groups are hydrophilic, whereas the long fatty acids show hydrophobic properties. In biological systems, a large variety of phosphoglycerids is found, since there is a high variability with regard to the alcoholic group and the fatty acids.

The name of the phosphoglycerides is based on the alcoholic head groups:

- Phosphatidic acid, **PA** (no head group), i.e. PO**PA**
- Phosphatidylcholine, PC, i.e. PO**PC**
- Phosphatidylethanolamine, **PE**, i.e. PO**PE**
- Phosphatidylglycerol, PG, i.e. PO**PG**
- Phosphatidylinositol, PI
- Phosphatidylserine, PS, i.e. PO**PS**

The PO in the lipids mentioned above, is the abbreviation for 1-palmitoyl-2-oleol. For MD simulations, GPCRs are mainly embedded into POPC lipid bilayers (Ivanov et al. 2005; Filizola et al. 2006; Henin et al. 2006; Strasser et al. 2007). The structure of POPC is presented in Fig. 5.2. However, other lipid models, like DOPC (dioleoylphosphatidylcholine) are used (Goetz et al. 2011).

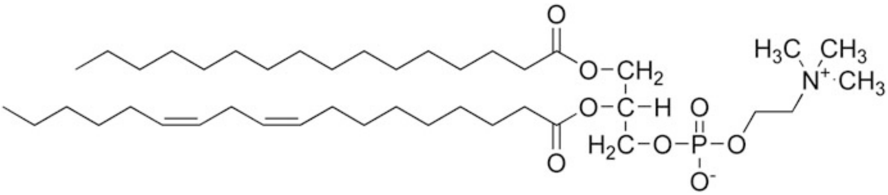
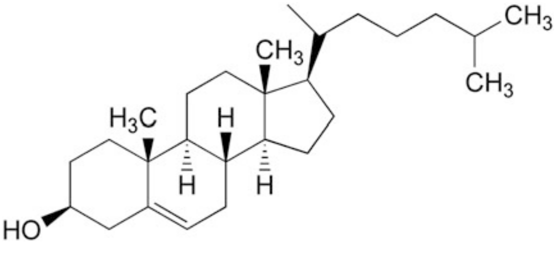


Fig. 5.2 Structure of POPC (1-palmitoyl-2-oleoylphosphatidylcholine)

Sterols are another important class of membrane lipids. One of the most prominent is the cholesterol (Fig. 5.3). The cholesterol scaffold contains four condensed rings leading to a distinct rigidity. This structure is hydrophobic and thus it is able to insert into the hydrophobic inner layer of the lipid bilayer. The polar hydroxyl moiety is located at the surface of the lipid bilayer.

Fig. 5.3 Structure of cholesterol



Cholesterol is sometimes found cocrystallized in combination with crystal structures of GPCRs. For example, a cholesterol specific binding site was identified for the human β_2 adrenergic receptor within the crystal structure 3D4S (Hanson et al. 2008).

5.2 Structure of the Phospholipid Bilayer

In Fig. 5.4, a site model of a lipid bilayer is presented. The hydrophobic chains point inside the lipid bilayer, whereas the polar head groups are facing towards the surrounding water.

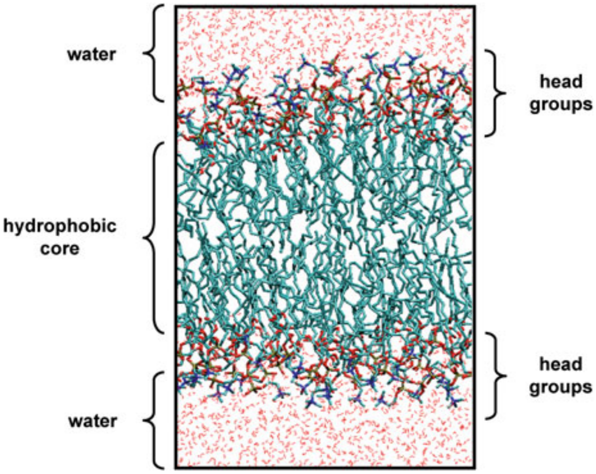


Fig. 5.4 Model of a lipid bilayer with water on both sides

5.3 Lipid Bilayer Models Used in Molecular Modelling

Several lipid models were constructed for use in molecular modelling. Some of them are summarized in Table 5.1.

Table 5.1 Summary of some lipids, often used in molecular modelling studies

DPPC	Dipalmitoylphosphatidylcholine
DMPC	Dimyristoylphosphatidylcholine
DOPC	Dioleoylphosphatidylcholine
POPC	1-palmitoyl-2-oleoylphosphatidylcholine
POPE	1-palmitoyl-2-oleoylphosphatidylethanolamine
PLPC	Palmitoyllinoleylphosphatidylcholine

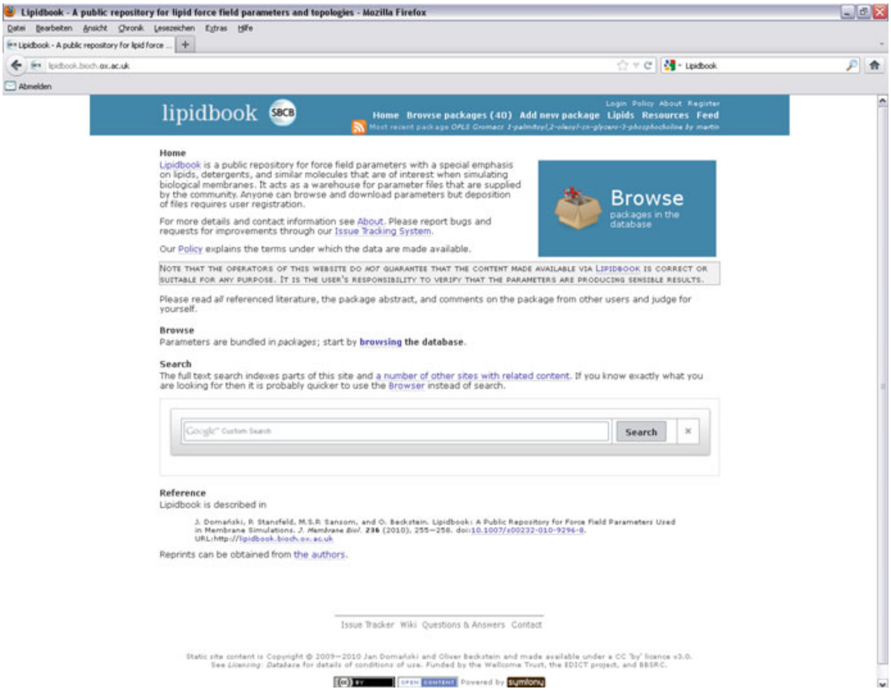
5.4 Internet Sources for Lipid Bilayer Models

In the internet, there are some sources which give a more detailed information with regard to lipid bilayers, including simulation parameters for GROMACS. At some sites in internet, equilibrated lipid bilayer models can be obtained via free download. A summary of the most important internet resources with regard to lipids is given in Table 5.2.

Table 5.2 Most important internet resources with regard to lipds

URL
http://lipidbook.bioch.ox.ac.uk
http://moose.bio.ucalgary.ca/index.php?page=Structures_and_Topologies
http://www.lrz-muenchen.de/~heller/membrane/membrane.html
http://www.scmbb.ulb.ac.be/Users/lensink/lipid/

A very comfortable site is lipidbook (http://lipidbook.bioch.ox.ac.uk) (Domanski et al. 2010). The aim of the lipidbook is "a public repository for force field parameters with special emphasis on lipids" (http://lipidbook.bioch.ox.ac.uk) (Fig. 5.5). Here, you can individually select the force-field, the parameter notation for distinct software and the kind of lipid (Fig. 5.6).


Lipidbook - A public repository for lipid force field parameters and topologies - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Lipidbook - A public repository for lipid force ...
lipidbook.bioch.ox.ac.uk
Lipidbook
Abmelden
lipidbook
SBCB
Login Policy About Register
Home Browse packages (40) Add new package Lipids Resources Feed
Most recent package OPLS Gromacs 1-palmitoyl,2-oleoyl-sn-glycero-3-phosphocholine by martin
Browse
packages in the database

Home

Lipidbook is a public repository for force field parameters with a special emphasis on lipids, detergents, and similar molecules that are of interest when simulating biological membranes. It acts as a warehouse for parameter files that are supplied by the community. Anyone can browse and download parameters but deposition of files requires user registration.

For more details and contact information see About. Please report bugs and requests for improvements through our Issue Tracking System.

Our Policy explains the terms under which the data are made available.

NOTE THAT THE OPERATORS OF THIS WEBSITE DO *NOT* GUARANTEE THAT THE CONTENT MADE AVAILABLE VIA LIPIDBOOK IS CORRECT OR SUITABLE FOR ANY PURPOSE. IT IS THE USER'S RESPONSIBILITY TO VERIFY THAT THE PARAMETERS ARE PRODUCING SENSIBLE RESULTS.

Please read *all* referenced literature, the package abstract, and comments on the package from other users and judge for yourself.

Browse

Parameters are bundled in *packages*; start by **browsing the database**.

Search

The full text search indexes parts of this site and a number of other sites with related content. If you know exactly what you are looking for then it is probably quicker to use the Browser instead of search.

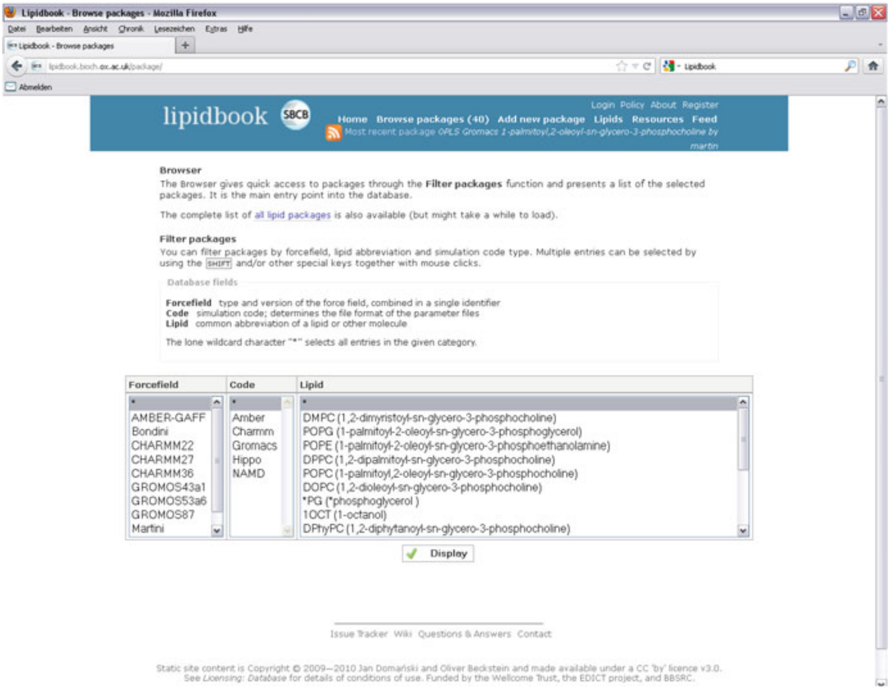
Reference

Lipidbook is described in

J. Domański, P. Stansfeld, M.S.P. Sansom, and O. Beckstein. Lipidbook: A Public Repository for Force Field Parameters Used in Membrane Simulations. *J. Membrane Biol.* **236** (2010), 255–258. doi:10.1007/s00232-010-9296-8. URL:http://lipidbook.bioch.ox.ac.uk

Reprints can be obtained from the authors.

Fig. 5.5 Starting page of lipidbook (http://lipidbook.bioch.ox.ac.uk). (Domanski et al. 2010)


Lipidbook - Browse packages - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Lipidbook - Browse packages
lipidbook.bioch.ox.ac.uk/package/
Lipidbook
Abmelden
lipidbook
SBCB
Login Policy About Register
Home Browse packages (40) Add new package Lipids Resources Feed
Most recent package OPLS Gromacs 1-palmitoyl,2-oleoyl-sn-glycero-3-phosphocholine by martin

Browser

The Browser gives quick access to packages through the **Filter packages** function and presents a list of the selected packages. It is the main entry point into the database.

The complete list of all lipid packages is also available (but might take a while to load).

Filter packages

You can filter packages by forcefield, lipid abbreviation and simulation code type. Multiple entries can be selected by using the SHIFT and/or other special keys together with mouse clicks.

Database fields

Forcefield type and version of the force field, combined in a single identifier
Code simulation code; determines the file format of the parameter files
Lipid common abbreviation of a lipid or other molecule

The lone wildcard character "*" selects all entries in the given category.

Forcefield	Code	Lipid
*	*	*
AMBER-GAFF	Amber	DMPC (1,2-dimyristoyl-sn-glycero-3-phosphocholine)
Bondini	Charmm	POPG (1-palmitoyl-2-oleoyl-sn-glycero-3-phosphoglycerol)
CHARMM22	Gromacs	POPE (1-palmitoyl-2-oleoyl-sn-glycero-3-phosphoethanolamine)
CHARMM27	Hippo	DPPC (1,2-dipalmitoyl-sn-glycero-3-phosphocholine)
CHARMM36	NAMD	POPC (1-palmitoyl,2-oleoyl-sn-glycero-3-phosphocholine)
GROMOS43a1		DOPC (1,2-dioleoyl-sn-glycero-3-phosphocholine)
GROMOS53a6		*PG (*phosphoglycerol)
GROMOS87		1OCT (1-octanol)
Martini		DPhyPC (1,2-diphytanoyl-sn-glycero-3-phosphocholine)

Fig. 5.6 Browser in lipidbook (http://lipidbook.bioch.ox.ac.uk). (Domanski et al. 2010)

5.5 Embedding a GPCR into a Lipid Bilayer

For embedding a GPCR into a lipid bilayer, different strategies are available. The most time consuming would be to simulate the whole system *de novo*, by putting an appropriate number of lipid and water molecules randomly around the GPCR and start a molecular dynamics simulation. Since this procedure is really time consuming, alternative methods are suggested: One approach could be to set an appropriate number of lipid molecules in appropriate orientation around the protein (Woolf and Roux 1996; Belohorcova et al. 1997). However, for this strategy, you must have access to an appropriate software, or you have to establish the software by yourself. Alternatively, for setting up your simulation box, you can start with already prepared lipid bilayers. Therefore, you can look at the mentioned internet resources (Table 5.2), download an equilibrated lipid bilayer model and use this for further calculations. Alternatively, you can construct a lipid bilayer individually with a distinct width with an appropriate software. One suitable software is `vmd` (http://www.ks.uiuc.edu/Research/vmd/), combined with some scripts, as described in more detail later on. The great advantage of the latter strategy is that you can individually adopt the size of your lipid bilayer with regard to the size of the GPCR or the GPCR-Gαβγ-complex. In this context you have to take into account two considerations: What do you want to simulate: Only a GPCR or a whole GPCR-Gαβγ-complex. Due to the larger size of a GPCR-Gαβγ-complex, compared to a GPCR, the lipid bilayer has to be large in case of a GPCR-Gαβγ-complex. However, in both cases, the lipid bilayer must be large enough in order to guarantee that the GPCR or GPCR-Gαβγ-complex is embedded well. This means, you should have a lipid bilayer with a width of optimally 1.0–1.5 nm around your protein. This guarantees that there are not undesirable interactions between proteins of virtual simulation boxes as a result of periodic boundary conditions, as illustrated in Fig. 5.7.

A lipid bilayer shell larger than 1.5 nm can be principally used, but this would not lead to any advantage, instead, the great disadvantage will be an exponential increase in simulation time. For simulation of a GPCR without the G protein, a width of the lipid bilayer of about 9–10 nm is recommended. Thus, in the first step you have to generate your lipid bilayer with an appropriate width (Fig. 5.8, step 1).

Subsequently, the GPCR has to be aligned into the lipid bilayer. A very good description in combination with the software vmd (http://www.ks.uiuc.edu/Research/vmd/) is found at the following internet site: http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/. A detailed description is given at the mentioned site. However, in the following, a short description of a slightly modified procedure using the script `combine.tcl`, available at (http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/) is presented. For the following procedure you need the shell script `vmd2gro`, which is shown later on. The script `vmd2gro` was tested in combination with `vmd 1.8.7`. Be aware, that in the presented version of `vmd2gro` the POPC molecules in vmd-notation are transferd into the POPC-notation used by Moose (http://moose.bio.ucalgary.ca/index.php?page=Structures_and_Topologies). Thus, for further use with GROMACS, you need the files `lipid.itp` and `popc.itp`. Both are available at http://moose.bio.ucalgary.ca/index.php?page=

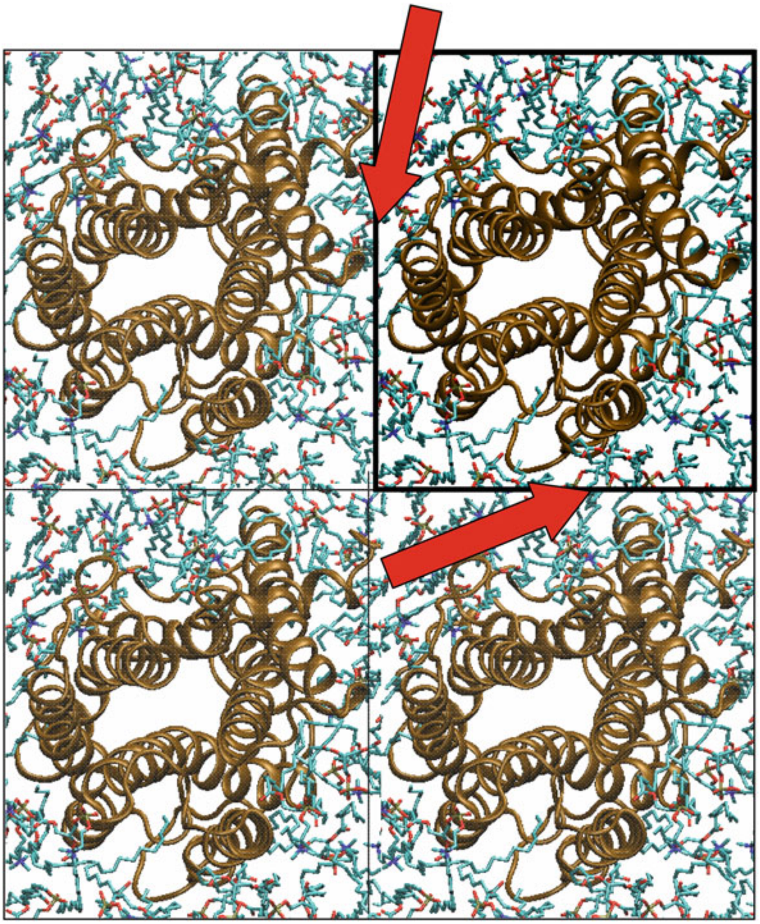


Fig. 5.7 Artificial close contacts of the protein in the simulation box with the proteins in the neighboured virtual simulation boxes due to periodic boundary conditions

Structures_and_Topologies and can also be found in the appendix (POPC Parameters). In the script `vmd2gro` the tcl-script `combine.tcl`, developed by Balabin and published at http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/ was used (see lines 192–276 in `vmd2gro` shown below). Before starting, you need your protein as a **correct** pdb-file, which can be used as an input file for the GROMACS command `pdb2gmx` without special options. In this example, this file is named `protein.pdb`. In the next step, you can start `vmd` to generate the lipid bilayer. Therefore, choose in the main menu `Extentions` → `Modelling` → `Membrane Builder`. There, choose `POPC` as lipid, because the script `vmd2gro` only considers `POPC` in the version presented here. Define the length of the lipid bilayer in x- and y-direction. Be aware to define the values in Å. In the actual version of the script `vmd2gro` a box size of about (10 × 10 × 10) nm is defined (see line 188).

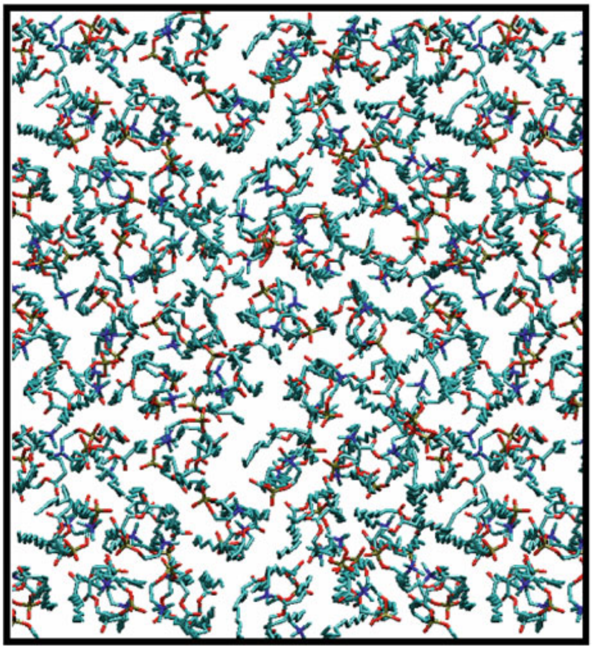


Fig. 5.8 Step 1: Generate or download a lipid bilayer of appropriate size

However you can change the values in the script of later on in the `gro`-file, if necessary. So, for example type 100 for width in x- and y-direction in the appropriate field of vmd. As "`ouput prefix`" type `membrane`. Subsequently, the membrane is generated and is shown in `vmd`. Additionally, in the directory, where `vmd` is started from, two files, named `membrane.pdb` and `membrane.psf` are generated. The extension `psf` means "`protein structure file`". In the next step, the file `protein.pdb` has to be loaded via `File` → `New Molecule`. To simplify the alignment of the protein in the lipid bilayer, use the menu `Graphics` → `Representations` → "`Coloring Method` → `Color ID` → `yellow`" and "`Drawing Method` → `New Cartoon`". Now, the coordinates of the protein have to be changed via `Mouse` → `Move` → `Molecule`. Be careful and move **ONLY** the protein and **NOT** the membrane. Moving the membrane in the "`move`" mode would result in a failure of the alignment! In the "`move`" mode, use the left mouse button for translation, the shift-button and the left mouse button for rotation around the z-axis and the shift-button in combination with the middle mouse button to rotate around the axis vertical to the screen. Therefore, click directly onto the protein with the mouse cursor. To leave the "`move`" mode, type "`r`" or use the menu button `Mouse` → `Rotate`. In the "`rotate`" mode, you can rotate the whole system (membrane and protein) as appropriate without changing any coordinates. In the next step, you have to use the "*move*" and "*rotate*" mode alternately to align the protein into the lipid bilayer. This procedure should be performed very carefully with regard to the placement of the GPCR in the membrane. Additionally, this procedure needs some practice and may take some time. If the alignment procedure is finished, the protein with the new coordinates has to be saved as pdb- and psf-file in the following manner: Use the menu buttons Extensions → Modelling → Automatic PSF Builder and a new window, opens: Step 1: In the field "Output basename", we write protein_autopsf for example and click onto the button "Load input files". Step 2: Choose "Everything" and click onto "Guess and split chains using current selections". Step 3: Click onto "Create chains". Step 4: Click onto "Apply patches and finish PSF/ PDB". An additional window opens, there, click "OK" and finish by clicking onto the button "Reset Autopsf". If you look into the directory, where vmd is started from, three new files protein_autopsf.log, protein_autopsf.pdb and protein_autopsf.psf are generated by the procedure, mentioned above. Now, vmd can be closed and the shell script vmd2gro can be startet:

```
> vmd2gro ↵
```

Subsequently, you have to define some basenames of files:

```
> Basename of membrane file: membrane ↵
> Basename of aligned protein file: protein_autopsf ↵
> Basename of output file (protein+membrane): temp ↵
```

The first two basenames have to be the same as used in the alignment procedure mentioned above. The third can have any basename, since these will be temporary files, which will be deleted automatically.

After that, the script performs some calculations and then it stops in order to ask you, if protonation states of amino acids should be changed. Here answer "*no*" by typing a "n". Subsequently, the command pdb2gmx is called within vmd2gro and you are asked to choose an appropriate force-field. For example, type "4". Now, vmd2gro performs some time-consuming calculations, like generation of a topology file. After some minutes, vmd2gro should have finished. Now, you should have some new files in your current working directory: membrane.gro, protein_autopsf.gro, protein_autopsf.gro, protein_autopsf.top and posre.itp. The files membrane.gro and protein_autopsf.gro contain the coordinates, relevant for the further steps. In membrane.gro, the POPC-lipid-bilayer with a hole and in an appropriate site-notation is given. Be aware, that the POPC in this notation can only be used with the parameters available in internet (http://moose.bio. ucalgary.ca/index.php?page=Structures_and_Topologies) and shown explicitly in the appendix (POPC Parameters). The file protein_autopsf.gro contains the coordinates of the aligned protein and the file protein_autopsf.top is the corresponding topology-file. The coordinates of both gro-files, membrane.gro and protein_autopsf.gro can be combined within one gro-file, containing now the protein and lipids. To do so, one can use the following LINUX-command-sequence:

```
> set nr_prot = `wc -l protein_autopsf.gro |
  cut -d' ' -f1` ↵
> set nr_mem = `wc -l membrane.gro | cut -d' ' -f1` ↵
> @ all_sites = ${nr_prot} + ${nr_mem} - 6 ↵
> echo "Protein in lipid bilayer" > prot_lip.gro ↵
> echo "$all_sites" >> prot_lip.gro ↵
> tail -n +3 protein_autopsf.gro |
  head -n -1 >> prot_lip.gro ↵
> tail -n +3 membrane.gro >> protein_lip.gro ↵
```

Now, you should have your protein and the lipid bilayer in the file protein_lip.gro. Of course, you can do the analogue manipulations manually with an editor. In protein_lip.gro, the lipid sites start again with number 1. To obtain a subsequent numbering and centring the structure in the simulation box, use the GROMACS command editconf:

```
> editconf –f protein_lip.gro -c –o system.gro ↵
```

The user of vmd2gro and the tcl-script combine.tcl developed by Balabin and published at http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/ (see lines 192–276 in vmd2gro shown below) should add the absolute path for top_all27_prot_lipid.inp in line 207. The script vmd2gro is shown in the following:

```
 1 #!/bin/tcsh
 2
 3 # vmd2gro: tcsh script to convert vmd format to
   gro format
 4 # for a lipid membrane/protein complex
 5
 6 # Detect and remove collisions between protein and
   membrane using a Tcl script for VMD
 7 # script statements follow a line beginning with
   the the pattern "# START_TCL"
 8
 9 # Variables containing file names: mem, prot, out
10 # Tcl scriptfile "out.tcl" will be created on the
   fly
11
12 echo -n "Basename of membrane file: "
13 set mem = "$<"
14
15 if (! -e "${mem}.pdb") then
16   echo "Missing pdb file: ${mem}.pdb"
17   exit 1
18 else if (! -e "${mem}.psf") then
19   echo "Missing psf file: ${mem}.psf"
20   exit 1
```

```
endif

echo -n "Basename of aligned protein file: "
set prot = "$<"

if (! -e "${prot}.pdb") then
  echo "Missing pdb file: ${prot}.pdb"
  exit 1
else if (! -e "${prot}.psf") then
  echo "Missing psf file: ${prot}.psf"
  exit 1
endif

echo -n "Basename of output file (protein+membrane): "
set out = "$<"

if (-e "${out}.pdb") then
  echo "File ${out}.pdb exists!"
  echo "Rename existing file or choose new file name and start again"
  exit 1
else if (-e "${out}.psf") then
echo "File ${out}.psf exists!"
echo "Rename existing file or choose new file name and start again"
  exit 1
endif

# Create and start tcl script: out.tcl

if (-e "${out}.tcl") then
echo "Tcl script file ${out}.tcl exists! Remove or rename it!"
  exit 1
endif

# Substitutions for MEM, PROT and OUT in Tcl script part

set begin_tcl = `grep -n "^# START_TCL" $0|cut -d ':' -f1`
echo $begin_tcl

tail -n +$begin_tcl $0|sed -e "s/MEM/$mem/"
```

```
-e "s/PROT/$prot/" -e "s/OUT/$out/"
> "${out}.tcl"

if ("$status") then
echo "Error creating Tcl script file
${out}.tcl! Terminating!"
exit 1
endif

# Starting Tcl script from working directory!

chmod u+x "${out}.tcl"

vmd -e ${out}.tcl -dispdev text

if ("$status") then
  echo "${out}.tcl failed! Terminating
calculations!"
exit 1
endif

echo "Output files ${out}.pdb and ${out}.psf
successfully created!"

echo "Going to convert pdb format to gro format"

# Removing obsolete files: pdb and psf files
of membrane and protein; tcl script file;
output psf file

rm "${mem}".{pdb, psf} "${prot}".{pdb, psf}
"${out}".tcl "${out}".psf

# Extracting protein and membrane structures
from out (*.pdb) file, deleting all water
molecules

grep 'POPC' "${out}.pdb"> "${mem}.pdb"
grep -v ’TIP\|POPC’ "${out}.pdb"|grep 'ATOM'|
 sed -e ’s/HSD/HIS/’> "${prot}.pdb"

rm "${out}.pdb"

# Converting pdb file for protein and possibly
change protonation state
```

```

echo "*********************************************************************************************"
echo -n "Do you want to modify the protonation state of amino acid ARG, ASP, GLU, HIS or LYS? (y/n): "
echo "*********************************************************************************************"

set answer = "$<"
set answer = `echo $answer|tr 'y' 'Y'`

set ARG = ""
set ASP = ""
set GLU = ""
set HIS = ""
set LYS = ""

if ("$answer" == "Y") then
  echo -n "Change protonation state of ARG? (y/n): "
set h = "$<"
set h = `echo $h|tr 'y' 'Y'`
if ("$h" == "Y") then
  set ARG = "-arg"
endif

echo -n "Change protonation state of ASP? (y/n): "
set h = "$<"
set h = `echo $h|tr 'y' 'Y'`
if ("$h" == "Y") then
  set ASP = "-asp"
endif

echo -n "Change protonation state of GLU? (y/n): "
set h = "$<"
set h = `echo $h|tr 'y' 'Y'`
if ("$h" == "Y") then
  set GLU = "-glu"
endif

echo -n "Change protonation state of HIS? (y/n): "
set h = "$<"
set h = `echo $h|tr 'y' 'Y'`
if ("$h" == "Y") then
```

```
  set HIS = "-his"
endif

echo -n "Change protonation state of LYS? (y/n): "
set h = "$ < "
set h = `echo $h|tr 'y' 'Y'`
if ("$h"=="Y") then
  set LYS = "-lys"
endif
endif

pdb2gmx -f "${prot}.pdb" -o "${prot}.gro" -p "${prot}.top" -ignh $ARG $ASP $GLU $HIS $LYS

rm "${prot}.pdb"

# Converting pdb file for membrane ****** *****************************************

setenv LC_NUMERIC '.'

# Initializations -map VMD atomic numbers to GRO atomic numbers (index)

set map = (4 3 5 1 2 17 23 20 21 22 24 25 28 30 31 32 33 45 48 51 54 57 60 63 65 67 70 73 76 79 82 36 39 40 41 42 92 95 98 101 104 107 110 113 116 119 122 125 128 131 85 88)

set gro_label = (C1 C2 C3 N4 C5 C6 O7 P8 O9 O10 O11 C12 C13 O14 C15 O16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 O33 C34 O35 C36 C37 C38 C39 C40 C41 C42 C43 C44 C45 C46 C47 C48 C49 C50 CA1 CA2)

set n_gro = $#map # number of sites per lipid molecule in gro notation
set n_pdb = 134 # number of atoms per lipid molecule in VMD notation

set gro_file = "${mem}.gro"

# Caluclations and mapping

```

```
@ units = ’wc -l "${mem}.pdb" | cut -d ' ' -f1’ /
$n_pdb # number of lipid molecules
@ number_of_atoms = $units * $n_gro

echo "Lipid membrane" >> $gro_file
echo "$number_of_atoms" >> $gro_file

set unit = 1
set atom_no = 1
while ("$unit"<="$units")
  set n = 1
  @ i1 = ($unit - 1) * $n_pdb
  while ("$n"<="$n_gro")
    @ pdb_line = $i1 + $map[$n]

    gawk -v u=$unit -v line=$pdb_line
-v atom=$atom_no -v label=$gro_label[$n]
’NR==line {printf("%5i%3s%7s%5i%8.3f%8.3f%8.3f\n",
u,"POP",label, atom,$6/10.0,$7/10.0,$8/10.0)}‘
"${mem}.pdb" >> $gro_file

    @ n++
   @ atom_no++
  end

  @ unit++
end

echo " 10.00 10.00 10.00" >> $gro_file

exit 0

#**************************************************
# START_TCL script part; do not edit or delete
this label! ***
# *** Following tcl commands for VMD ***
# embed (parts of) protein into a membrane
# Ilya Balabin (ilya@ks.uiuc.edu), 2002–2003
#
# You need: a) membrane structure
(membrane.psf/pdb);
# b) properly oriented and aligned to the membrane
# protein structure (protein.psf/pdb)


# set echo on for debugging
echo on

# need psfgen module and topology
package require psfgen
topology top_all27_prot_lipid.inp

# load structures
resetpsf
readpsf MEM.psf
coordpdb MEM.pdb
#readpsf protein.psf
readpsf PROT.psf
#coordpdb protein_aligned.pdb
coordpdb PROT.pdb

# can delete some protein segments; list them in brackets on next line
set pseg2del { }
foreach seg $pseg2del {
  delatom $seg
}

# write temporary structure
set temp "temp"
writepsf $temp.psf
writepdb $temp.pdb

# reload full structure (do NOT resetpsf!)
mol load psf $temp.psf pdb $temp.pdb

# select and delete lipids that overlap protein:
# any atom to any atom distance under 0.8A
# (alternative: heavy atom to heavy atom distance under 1.3A)
set sellip [atomselect top "resname POPC"]
set lseglist [lsort -unique [$sellip get segid]]
foreach lseg $lseglist {
# find lipid backbone atoms
set selover [atomselect top "segid $lseg and within 0.8 of protein"]
# delete these residues
set resover [lsort -unique [$selover get resid]]
foreach res $resover {
  delatom $lseg $res
}
}

# delete lipids that stick into gaps in protein
foreach res { } {delatom $LIP1 $res}
foreach res { } {delatom $LIP2 $res}

# delete lipids that fall out of the PBC box
# the following numbers are for example only; yours are different!
set xmin -55
set xmax 41
set ymin -51
set ymax 34
foreach lseg {"LIP1" "LIP2"} {
# find lipid backbone atoms
set selover [atomselect top "segid $lseg and (x<$xmin or x>$xmax or y<$ymin or y>$ymax)"]
# delete these residues
set resover [lsort -unique [$selover get resid]]
foreach res $resover {
  delatom $lseg $res
}
}

# write full structure
writepsf OUT.psf
writepdb OUT.pdb

# clean up
file delete $temp.psf
file delete $temp.pdb

# non-interactive script
quit
```

In Fig. 5.9, the membrane with a hole, created as described above, is shown. The file `membrane.gro` should look similar, if loaded into `vmd`.

After establishing an appropriate hole in the lipid bilayer and putting the GPCR into the hole, you should receive a system as shown in Fig. 5.10. The file `protein_lip.gro`, created above, should look similar. As you can see in the figure, there is a significant gap between the lipid bilayer and the GPCR.

Now, the system consists of the lipid bilayer and the GPCR. Using the GROMACS commands `grompp` and `mdrun`, the system can be minimized (see Chap. 6). Thus,

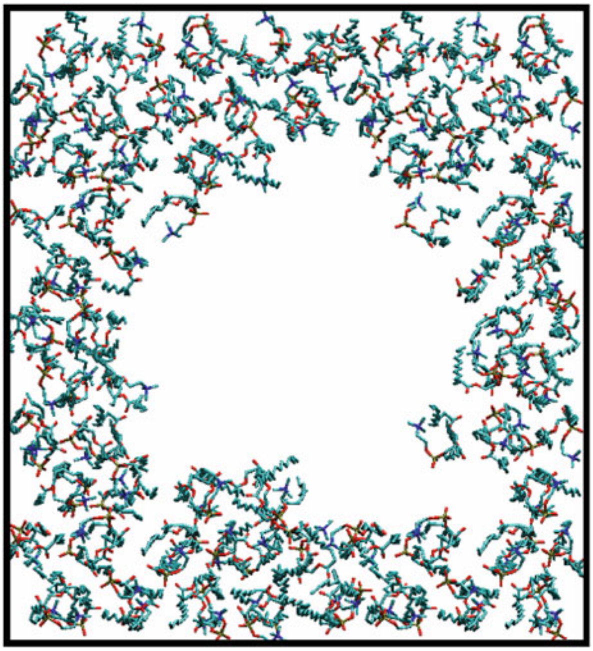


Fig. 5.9 Step 2: Generate a hole of appropriate size for the GPCR in the lipid bilayer

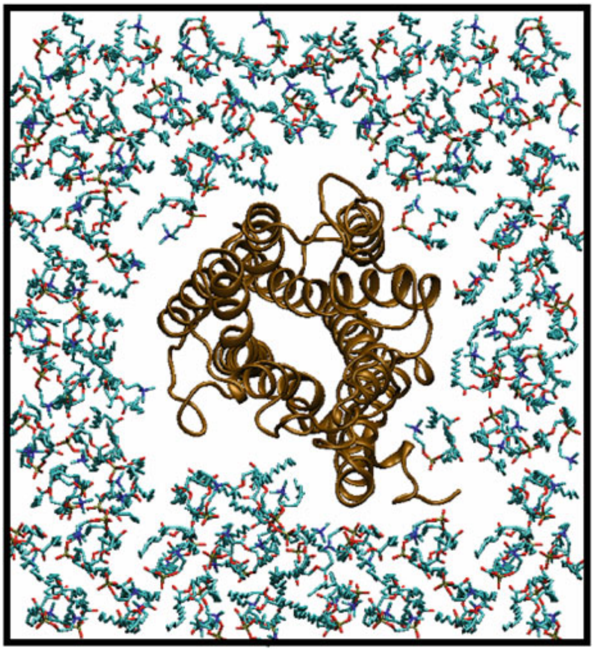


Fig. 5.10 Step 3: Placement of the GPCR or GPCR-G-protein-complex in the lipid bilayer

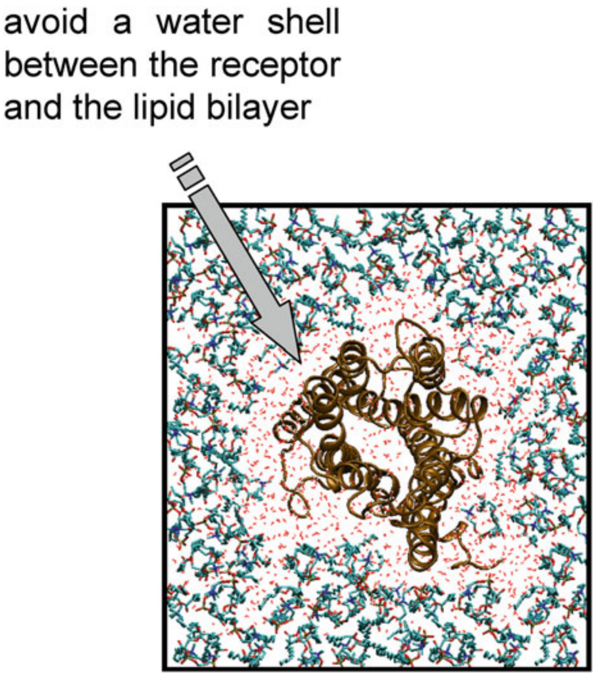


Fig. 5.11 An artificial water shell between the GPCR and the lipid bilayer, as consequence of wrong system setup

one might think, that the system has to be solvated in the next step. Doing so would lead to an artificial system, as pointed out in Fig. 5.11.

Due to the gap between the GPCR and the lipid bilayer, a large number of water molecules would be put into this gap during the solvation of the system. This water between the GPCR and the lipid bilayer is artificial and may lead to problems during the simulation or to artefacts, because the hydrophobic transmembrane domains of the receptor and the hydrophobic fatty acid side chains of the lipids are in contact to the hydrophilic water. Thus, both, the hydrophobic side chains and lipids might obtain energetically more favoured conformations without contact to the hydrophilic water. This may lead to instabilities of the receptor during simulation. However, some 10 water molecules all in all between lipid and receptor should not lead to problems during the simulation. They can be removed, but in most cases, they move into the extra- or intracellular water during the simulation. In order to avoid scenarios, as illustrated in Fig. 5.11, the lipid bilayer should be equilibrated around the GPCR (Fig. 5.12) before solvating the system. Therefore, different simulation protocols can be used. However, positions constraints have to be put at least onto the protein in order to avoid any conformational change of the protein during the lipid-equilibration process. In order to obtain an equilibration of the lipids in the xy-plain, slight position constraints might be put onto the z-coordinates of the lipids. In general, the modeller is encouraged to perform some different equilibration protocols in order to obtain an optimal structure. After this equilibration step, the lipid bilayer is fitted well to the GPCR and the gap between the GPCR and the lipid bilayer is removed.

Now, the system can be solvated in the next step. An optimally solvated box should look, as shown in Fig. 5.13.

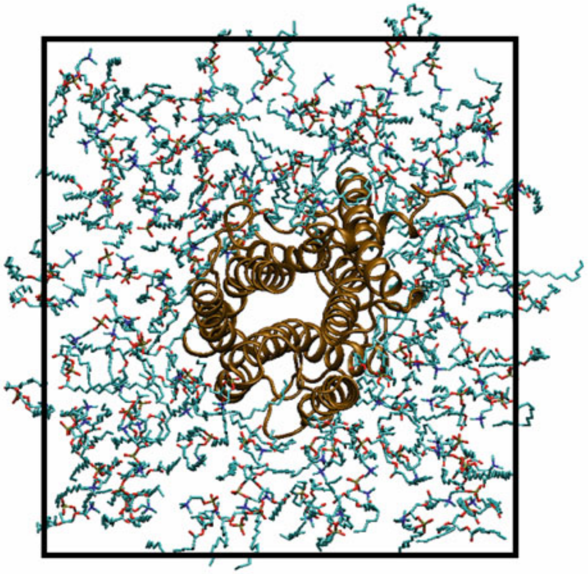


Fig. 5.12 Step 4: Equilibration of the lipid bilayer around the GPCR

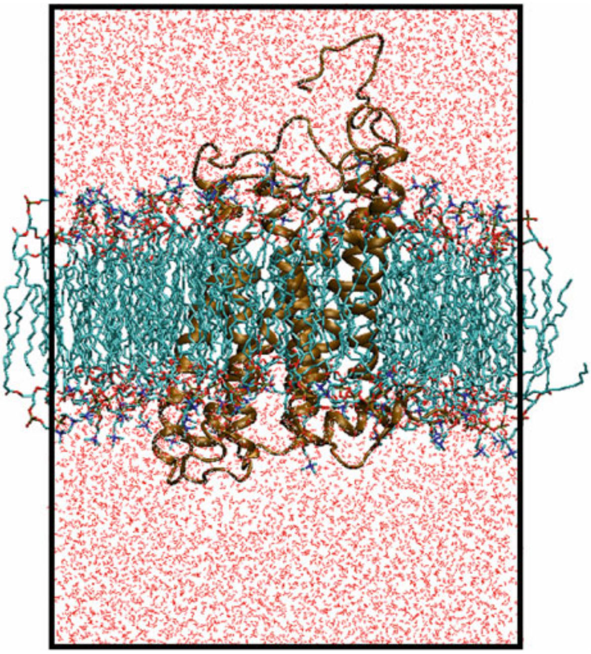


Fig. 5.13 Step 5: A well prepared simulation box, containing the GPCR, the lipid bilayer and extra- and intracellular water

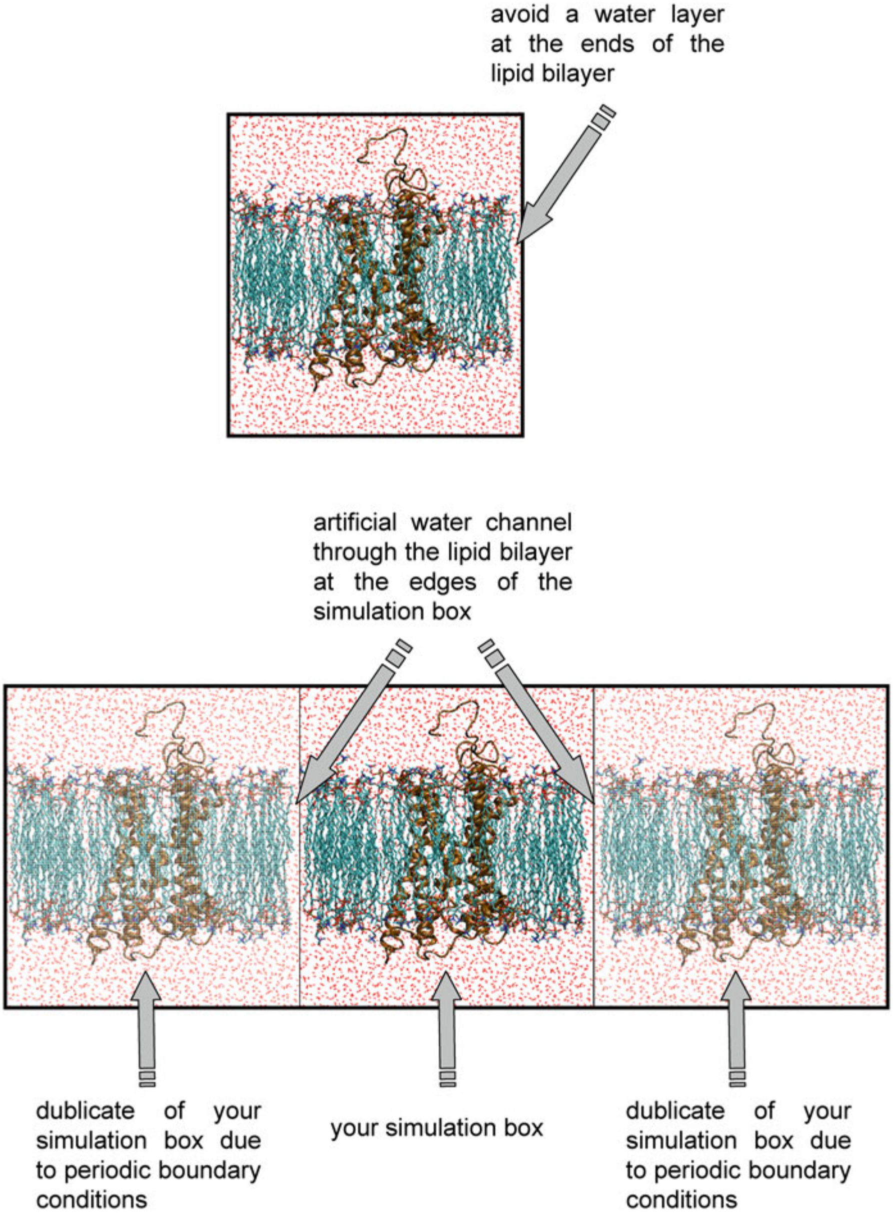

avoid a water layer at the ends of the lipid bilayer
artificial water channel through the lipid bilayer at the edges of the simulation box
dublicate of your simulation box due to periodic boundary conditions
your simulation box
dublicate of your simulation box due to periodic boundary conditions

Fig. 5.14 Artificial water channels through the lipid bilayer at the edges of the simulation box as consequence of wrong system setup

With regard to solvation you have to look carefully onto the size of your simulation box: If the defined box size is a little bit larger, than the width of the lipid bilayer, and you perform the solvation, you may get another artefact, as shown in Fig. 5.14. Here, water channels/layers through the lipid bilayer are established. This is a completely wrong artefact and you should not do simulations with such systems. If you detect such a water channel/layer after solvation, you may remove the solvent, decrease the box size in an appropriate manner and solvate again. These steps should be repeated until the water channel/layer through the lipid bilayer is no longer observed.

After solvation, the system should be minimized using the GROMACS command `grompp` and `mdrun` (see Chap. 6). In the last step, the system has to be neutralized (see Chap. 6).

In the following box a short, stepwise summary of the alignment of a GPCR in the lipid bilayer is given.

□ Construct a lipid bilayer or obtain it via download of a server
□ Align your GPCR correctly into the lipid bilayer
□ Remove the lipid molecules which overlap with the GPCR
□ Center the system in the simulation box
□ Minimize the system with GROMACS
□ Equilibrate the lipids around the GPCR, position restraints should be put onto all sites of the protein using appropriate GROMACS commands
□ Solvate your lipid-GPCR-complex with water in an appropriate manner (see also Chap. 6)
□ Minimize the simulation box
□ Neutralize the system and minimize again (see also Chap. 6)