g_membed: Efficient Insertion of a Membrane Protein into an Equilibrated Lipid Bilayer with Minimal Perturbation

MAARTEN G. WOLF,[1] MARTIN HOEFLING,[2] CAMILO APONTE-SANTAMARÍA,[2] HELMUT GRUBMÜLLER,[2] GERRIT GROENHOF[1]

[1]*Computational Biomolecular Chemistry Group, Max-Planck-Institute for Biophysical Chemistry, Göttingen D-37077, Germany*
[2]*Department for Theoretical and Computational Biophysics, Max-Planck-Institute for Biophysical Chemistry, Göttingen D-37077, Germany*

Abstract: To efficiently insert a protein into an equilibrated and fully hydrated membrane with minimal membrane perturbation we present a computational tool, called g_membed, which is part of the Gromacs suite of programs. The input consists of an equilibrated membrane system, either flat or curved, and a protein structure in the right position and orientation with respect to the lipid bilayer. g_membed first decreases the width of the protein in the xy-plane and removes all molecules (generally lipids and waters) that overlap with the narrowed protein. Then the protein is grown back to its full size in a short molecular dynamics simulation (typically 1000 steps), thereby pushing the lipids away to optimally accommodate the protein in the membrane. After embedding the protein in the membrane, both the lipid properties and the hydration layer are still close to equilibrium. Thus, only a short equilibration run (less then 1 ns in the cases tested) is required to re-equilibrate the membrane. Its simplicity makes g_membed very practical for use in scripting and high-throughput molecular dynamics simulations.

© 2010 Wiley Periodicals, Inc.   J Comput Chem 31: 2169–2174, 2010

Key words: computer simulations; membrane; Gromacs; molecular dynamics; protein insertion

Introduction

The number of computer simulations to probe the molecular details of membrane proteins is rapidly increasing.[1] Proteins embedded in or associated with membranes are of interest because they perform various crucial tasks, including (selective) transport, signal transduction and energy conversion. It is estimated that in most organisms 20–30% of the proteins encoded in the genome are membrane proteins[2] and that many drug targets are located at or near the cell surface (60% in 2006).[3] Nevertheless, these percentages are not reflected in the ratio of studies between soluble and membrane proteins. Because of the requirement of a lipid bilayer, membrane proteins are more complicated to crystallize or study with NMR. Consequently less than 1% of the structurally resolved proteins are membrane proteins.[4] However, the number of membrane protein structures in the pdb-database is rising quickly (see http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html for a maintained list) and thus their availability for computational studies.

Also on the simulation side, setting up a membrane protein system requires additional effort*s* compared to soluble proteins. First, an appropriate membrane model has to be selected for which parameters are available in the desired force field. Setting up plain membrane simulations can now be considered standard procedure[1] and hence will not be further discussed here. Furthermore, many equilibrated membrane patches and the corresponding force field parameters can be downloaded from webservers (URLs given in[1]) maintained by research groups specialized in membrane simulations. Second, and crucially, once a suitable membrane is selected the protein has to be embedded in the membrane before the production runs can be performed.

Several protocols have been suggested to insert a protein into a lipid bilayer, but up to now no standard has been established, mainly for reasons discussed later. The simplest way to insert a protein is to remove lipids and solvent molecules that overlap with the protein after combining the coordinate sets (X, Y, Z). However, the highly disordered nature of lipid tails results in an irregular and oversized hole and hence a poor lipid-protein packing. Long simulation runs are then needed to equilibrate the system. To obtain a good lipid-protein packing one can build a lipid bilayer around a protein from preequilibrated and prehydrated single lipids.[5] Unfortunately, assembling the bilayer from uncorrelated lipids will result

---
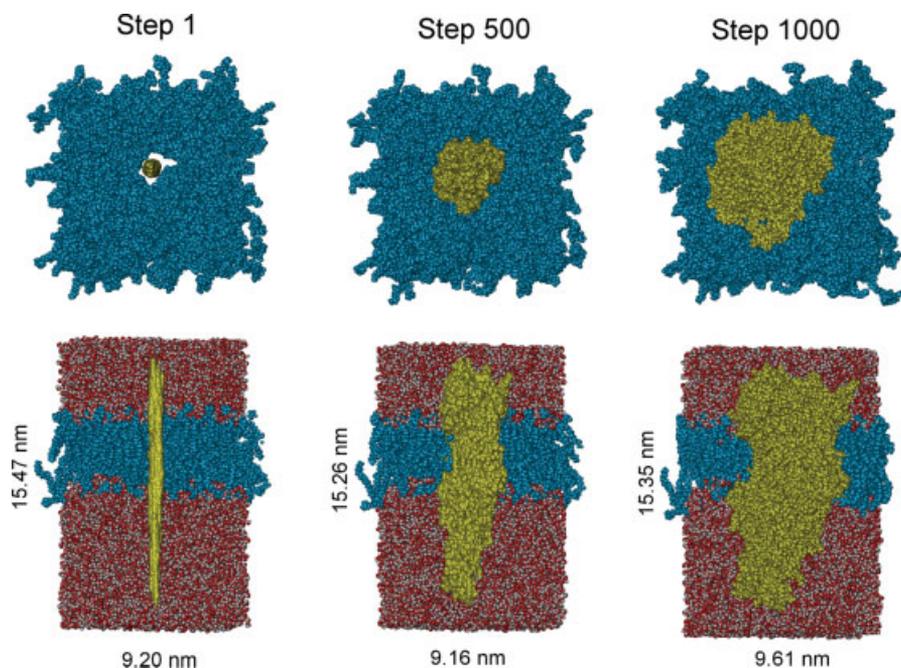
Correspondence to: M. G. Wolf; e-mail: mwolf@gwdg.de

Figure 1. The embedding proces. Three snapshot of a protein (Reaction Center, see Table 1) grown within the membrane using g_membed are shown in top view (upper) and side view (lower). The protein, membrane and water are displayed as Van der Waals spheres and colored yellow, blue, and red/white, respectively. The width of the protein in the membrane plane is 0.1, 0.55, and 1.0 for step 1, step 500 and step 1000, respectively. Also the length of the box-vectors are shown.

in bad contacts (overlapping atoms), which require extensive optimisation by, for instance, rigid body translation and rotations.[5] A more sophisticated method uses repulsive forces to create a hole at the intended protein position before inserting the protein itself.[6,7] This will result in a proper lipid-protein packing in an equilibrated membrane, but needs considerable tuning by the user to find the optimal hole generating parameters.[8] Recently, it has also been proposed to inflate the lipid bilayer, insert the protein and then, alternatingly deflate the lipid bilayer in small steps and perform an energy minimization, until the area per lipid reaches the reference equilibrium value.[8] However, this method cannot be applied with a solvated membrane and thus requires resolution. Furthermore, it is not clear if the membrane remains in equilibrium during the deflation procedure. The lipid tails that were entangled in the equilibrated membrane are likely to clash when going from a very diluted (inflated) to a fully packed membrane, resulting in a different packing of the lipid tails. Finally, with coarse grained simulations it is possible to let the lipids self-assemble around the membrane protein.[9] Unfortunately, the level of detail of coarse grained simulation might not be sufficient to assess the relevant interactions in membrane proteins adequately. Also the inverse mapping problem, i.e., obtaining the atom positions from the coarse grained coordinates, is highly nontrivial.

To address the issue more systematically, we propose that an insertion method needs to fulfil four requirements: (1) It should be easy to use, without the requirement of parameter tuning or manual intervention during or after the embedding process. (2) It should yield a structure that is close to equilibrium, to reduce the equilibration simulation time before production runs. (3) It should be easily automated, to allow for large scale high-throughput simulations. (4) It should be distributed as part of a popular MD package, to maintain availability, functionality and retraceability. None of the aforementioned methods satisfies these four criteria and therefore we developed a new Gromacs[10] tool, g_membed, that grows a protein into a lipid bilayer during a short md simulation (1000 steps typically). By using an already hydrated and equilibrated lipid bilayer the output system (protein embedded in the solvated membrane) only requires a short equilibration run (maximal 1 ns in our test systems) to reequilibrate the membrane part of the system. Finally, by making g_membed part of Gromacs availability, functionality, and retraceability is guaranteed.

Method

The basic idea is to slowly grow the protein into an already equilibrated membrane, thereby pushing away the lipids and waters and fit the protein nicely into the lipid bilayer. Starting point of the growth process is the narrowed protein, where the narrowing of the protein is performed within the membrane plane, but not in the direction of the membrane normal. In the limiting case, all protein atoms start on a line perpendicular to the membrane plane. Shrinking the protein in 2D instead of 3D is crucial, as it avoids severe perturbation of the lipids, especially when approaching the limiting cases (a line instead of a point). The growth phase itself is a short md run. After each md step the size of the protein is slightly increased until the protein has reached its initial size (see Fig. 1 for a typical example). During this process all protein–protein interactions, including bonded interactions, are switched off.

Table 1. Systems Used to Test g_membed.

	pdb-Code	Atoms (nr)	A_{prot} (nm^2)	Membrane
Integrin αIIbβ3[11]	2K9J	1413	5.78	popc
β-Barrel Platform[12,a]	2JMM	2372	9.52	popc
Reaction Center[13]	1PRC	22,196	29.59	dopc
Yeast Aquaporin[14]	2W2E	15,868	39.10	dopc
β-Barrel Platform[12,a]	2JMM	1584	7.25	pope (vesicle)

[a]The discrepancies are due to the use of different force fields.

The method, as implemented into the Gromacs tool g_membed (*embed*ding a protein within a *mem*brane) requires a file with the protein structure overlapping the lipid bilayer spanning the *xy*-plane at the desired position and orientation as input (we assume the user already has this knowledge about the system). Optimal placement of the protein with respect to the membrane will avoid needlessly-long equilibration runs after embedding. In the case of a curved bilayer (e.g., a vesicle), it is important that the protein is overlapping a membrane part that has its normal directed along the z-axis. g_membed then performs the following steps to embed the protein (or any other desired group of atoms defined as a Gromacs index group) within a lipid bilayer.

1. Protein narrowing—The coordinates of the atoms in the protein are scaled with respect to the geometrical center of the transmembrane part of the protein by a user specified scaling factor of the original (input) coordinates in the *xy*-plane and, if applicable, in the *z*-direction. Normally the protein should not be scaled in the *z*-direction. However in special cases, such as a protein that has the same height as the bilayer, increasing the size of the protein in the *z*-direction prevents lipids to envelop the protein during the growth phase.
2. Remove overlapping molecules—Every molecule not part of the protein for which at least one atom is within a user defined radius of a protein atom will be removed. If the difference between the number of lipids removed from the lower (n_{lower}) and the upper (n_{upper}) membrane leaflet is not equal to a user defined number (n_{diff}), i.e., $n_{lower} - n_{upper} \neq n_{diff}$, additional lipids will be removed, such that this equality will be obtained. This option is usefull when inserting an asymetrically shaped protein.
3. Growth phase—Iterate steps 3a and 3b $nxy + nz$ times to grow the narrowed protein to its original size.
   a. md step—Do a normal md step.
   b. Protein resizing—Change the atom coordinates of the protein by linear interpolation between the coordinates of the narrowed protein (step 1) and those of the input configuration by

$$\mathbf{r}_i = \mathbf{r}_{geom} + \mathbf{s}_i \cdot (\mathbf{r}_0 - \mathbf{r}_{geom}) \quad (1)$$

$$\mathbf{s}_i = \begin{cases} s_{i,x} = s_{i,y} = s_{0,xy} + \frac{i \cdot (1 - s_{0,xy})}{n_{xy}} \text{ and } s_{i,z} = s_{0,z} \\ \qquad\qquad\qquad\qquad\qquad\qquad i \leq n_{xy} \\ s_{i,x} = s_{i,y} = 1 \text{ and } s_{i,z} = s_{0,z} + \frac{(i - n_{xy}) \cdot (1 - s_{0,z})}{n_z} \\ \qquad\qquad\qquad\qquad\qquad n_{xy} < i \leq n_{xy} + n_z \end{cases} \quad (2)$$

with $\mathbf{r}_i$ the protein atom coordinates at step i, $\mathbf{r}_0$ the input atom coordinates of the protein, $\mathbf{r}_{geom}$ the coordinates of the

geometrical center of the transmembrane region, $\mathbf{s}_i$ the scaling factor at step i, $s_{0,xy}$, and $s_{0,z}$ the initial scaling factors and n_{xy} and n_z the number of embedding steps in the *xy*-plane and the *z*-direction, respectively. This way the atom coordinates of the proteins reach the input configuration in n_{xy} steps in the *xy*-plane and, if applicable, after the *xy*-dimension is completely expanded in n_z steps in the *z*-dimension.

The output will be a file containing the protein structure properly embedded in the membrane. All atom coordinates of the protein in the output file will be equal to the coordinates provided in the input (see the Appendix for a user manual).

Case Studies

As a test we have applied g_membed to insert four proteins of various shape and size: Integrin αIIbβ3 2K9J,[11] β-Barrel Platform 2JMM,[12] Reaction Center 1PRC,[13] and Yeast Aquaporin 2W2E[14] (see Table 1). We have embedded the first two into a united-atom (POPC[15,16]) and the latter two into an all-atom (DOPC[17]) membrane. In addition, we have embedded the β-Barrel Platform within the lipid bilayer of a vesicle (POPE), see Figure 2. The crystal waters available in the Reaction Centers pdb entry were preserved and considered as part of the protein. Initially, the width of the protein in these tests was scaled to 10% of the full-size protein ($s_{0,xy} = 0.1$). All proteins listed in Table 1 were inserted successfully within the bilayer within 1000 md steps.

To assess the time required to reach equilibrium after embedding the protein within the membrane, a 10 ns standard md simulation was performed after the embedding (insertion within the vesicle not
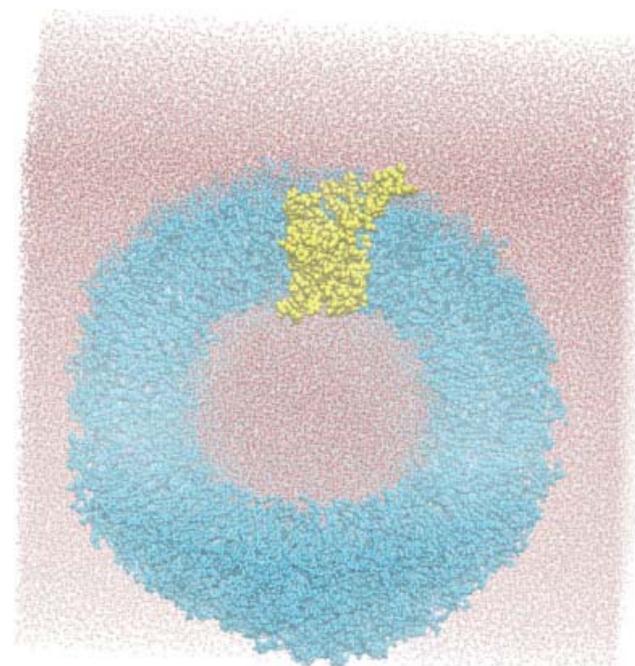

Figure 2. A protein embedded in a vesicle bilayer. The β-Barrel platform (yellow) has been embedded within a POPE vesicle (blue) in water (red/white) with g_membed.
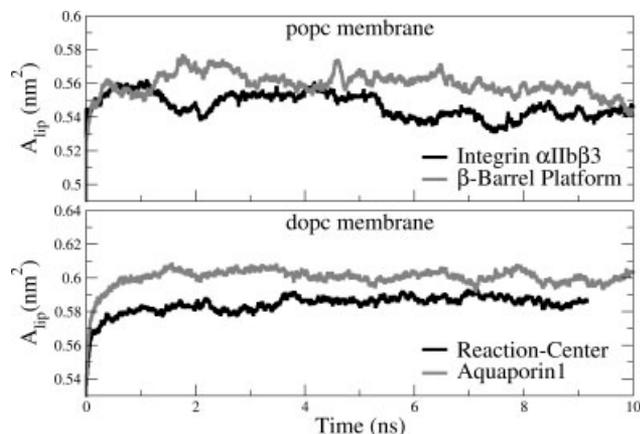
Figure 3. Equilibrating a system after embedding a protein with g_membed into a membrane starting with an initial scaling factor of 0.1. The area per lipid (A_{lip}, eq. 3) as a function of simulation time shows that within 1 ns the equilibrium area per lipid is reached.

included). Figure 3 shows the area per lipid (A_{lip}) as a function of time, where A_{lip} is obtained through

$$A_{lip} = 2 * (A_{xy} - A_{prot})/n_{lip} \quad (3)$$

with n_{lip} the number of lipids in the membrane, A_{xy} the area spanned by the *x*- and *y*-box vectors (membrane plane) and A_{prot} the protein area estimated by projecting all the atoms of the protein that overlap the membrane slab on a two-dimensional square grid in the *xy*-plane and summing the area of all grid elements that contain a protein atom. From Figure 3 it is clear that already within 1 ns the system reaches the equilibrium area per lipid. For a pure popc and dopc membrane system we found an area per lipid of 0.59 and 0.62, respectively, which is slightly higher than the values we found for the protein-membrane system. This discrepancy can partly be explained by the fact that the protein area is overestimated, resulting in an underestimation of the area per lipid. Furthermore, since the influence of proteins on lipid properties is still not completely understood, it is not clear whether the area per lipid should actually be unchanged after protein insertion.[18, 19]

Reaching the equilibrium area per lipid does not necessarily imply that the membrane is in equilibrium. As a more rigorous test, we therefore also analyzed the deuterium order parameter, S_{CD}, a measure for the amount of disorder in the lipid tails. Comparing the deuterium order parameter before and after embedding the protein into the membrane did not show any significant differences. In addition, we calculated the density profile of water normal to the membrane plane to compare the hydration of the membrane. Also here we found no significant difference before and after embedding the protein. Monitoring the deuterium order parameter and the water density profile normal to the membrane over time indicate that the membrane is still in equilibrium.

To determine the time required to reach area equilibrium, we calculated the relaxation time τ associated with the area per lipid by fitting

$$A_{lip} = \langle A_{lip} \rangle - A_0 * exp(-t/\tau) \quad (4)$$

to the area per lipid as a function of time (Fig. 3). In eq. 4, $\langle A_{lip} \rangle$ is the ensemble average of the area per lipid, where we averaged over the 2–10 ns time interval and A_0 the fitted difference with $\langle A_{lip} \rangle$ at $t = 0$. The resulting relaxation times are listed in Table 2, column 0.1. The average relaxation time within the scaling factor range 0.1–0.3 (including data not listed in Table 2) is 180 ± 190 ps. From this value and the values listed in Table 2 we expect that a 1 ns equilibration run suffices after embedding a protein in an equilibrated membrane using g_membed. Nevertheless, as the number of cases studied here is limited, the required equilibration time might differ for some specific systems.

Finally, we examined the influence of the protein scaling factor $s_{0,xy}$ used in the first step of the embedding process. We expect that a larger initial scaling factor implies slower equilibration, since more lipids have to be moved to accomodate the protein. We embedded a Yeast Aquaporin tetramer and the β-Barrel Platform starting with initial scaling factors ranging from 0.1 to 0.7, followed by a 10 ns standard simulation. The root mean square deviation (rmsd) of the Yeast Aquaporin tetramer with respect to the X-ray structure approaches the same value, 0.15 nm, for all initial scaling factors, whereas the rmsd for the β-Barrel Platform ranges from 0.3 to 0.6 nm without any correlation to the initial scaling factor. These findings indicate that the protein structure is unperturbed by the initial scaling factor. Table 2 shows the relaxation times of the area per lipid for different initial scaling factors. These values show that indeed our hypothesis that a smaller scaling factor implies slower equilibration is true. When the scaling factor approaches a value, for which the removed lipid area is only slightly smaller than the estimated protein area (0.6 and 0.7 for the β-Barrel Platform and Yeast Aquaporin, respectively) the perturbation to the membrane as measured by the area per lipid, deuterium order parameter and water density profile is minimal in the cases tested. The length of the simulation to equilibrate the system is then mainly determined by the equilibration of the protein.

Conclusions

We have presented a program, g_membed, for efficient insertion of a protein structure into a hydrated and equilibrated membrane. g_membed can handle both flat and curved membranes, thereby for instance also allowing embedding of protein structures within vesicle bilayers. The program only minimally perturbs the properties and the hydration of the system during protein insertion. As a consequence short subsequent equilibration runs suffice, enabling drastically increased throughput. In addition, one set of parameters (set as default) can be aplied to insert proteins of various shape and

Table 2. Relaxation Time τ (**ps**) of the Area per Lipid After Embedding the Protein in a Membrane with Different Initial Protein Sizes.

		Initial protein scaling factor $s_{0,xy}$						
		0.1	0.2	0.3	0.4	0.5	0.6	0.7
Protein (membrane)	Integrin αIIbβ3 (popc)	2						
	β-Barrel Platform (popc)	113	115	622	2		2	0
	Reaction Center (dopc)	306						
	Yeast Aquaporin (dopc)	153		66		49		0

size into an arbitrary lipid bilayer, suggesting this set to be useful for most proteins.

We consider the following two setups most useful. (1) The protein is embedded within a membrane slab that is relatively small compared to the protein size, so that a small number of lipids should be removed. Consequently, the initial scaling factor $s_{0,xy}$ needs to be set to a very small value (e.g., 0.1 of the full protein size). After embedding, a 1 ns equilibration is recommended to reequilibrate the membrane, but longer equilibration times may be required for the protein. This setup is the default for g_membed. (2) Alternatively, if the protein is to be embedded within a membrane slab that is large with respect to the proteins size, the protein scaling factor $s_{0,xy}$ can be set to a value, such that g_membed removes a lipid portion that is slightly smaller than the estimated protein area (depending on the protein size a scaling factor between 0.5 and 0.7). In this case, the perturbation of the system is even smaller, and the length of the equilibration run is determined by the time it takes to equilibrate the protein structure.

Computational Details

g_membed

As g_membed uses the molecular dynamics code implemented in Gromacs,[10] all system options available to a standard md simulation also apply to g_membed. The Integrin αIIbβ3 and the β-Barrel Platform were simulated using the OPLS-AA force field[20] and embedded in the popc lipid bilayer[15, 16] combined with simple point charge (SPC) water.[21] For the Reaction-Centre and Yeast Aquaporin we used the Amber03 parameter set[22] with the dopc membrane[17] and TIP3P water.[23] During embedding of the β-Barrel Platform within the pope vesicle we applied the Gromos-87 force field[24] with SPC water.

To insert a protein in the membrane by g_membed, 1000 time steps were performed. A Van der Waals cut-off of 1.4 nm was used and the electrostatic interactions were treated using PME[25] with a real space cut-off of 1.0 nm. All bonds were constraint using the LINCS algorithm,[26] allowing a time step of 2 fs. The temperature and pressure were kept constant at 300 K and 1 bar using velocity rescaling[27] and Berendsen semi-isotropic pressure coupling,[28] respectively. Protein–protein interactions were excluded and the atoms of the protein were set to the position specified in step 3 of the g_membed protocol.

Equilibration Runs

After embedding the protein in the membrane a 10 ns equilibration run was performed. The same system parameters were used as during embedding, with the exception that the temperature of the popc simulations was set to 323 K to prevent ordering of the bilayer. Also, the protein–protein interactions are not excluded and the atom positions of the protein are no longer fixed.

Acknowledgement

Appendix: Manual

The required input for g_membed is a tpr-file containing the group to embed (further called protein) and the (solvated) membrane. The protein should be placed at the desired position and orientation overlapping the lipid bilayer by using for instance pymol.[29] If the position and orientation of the protein is satisfactory, merge the protein and (solvated) membrane structure files into one (e.g., `merged.gro`) and create a matching topology file.

Download `sample.mdp` from our website (http://wwwuser.gwdg.de/~ggroenh/membed.html) or set the following options in an mdp file.

```
integrator     = md
energygrp      = Protein (or other group that you want to insert)
freezegrps     = Protein
freezedim      = Y Y Y
energygrp_excl = Protein Protein
```

Generate the input file for g_membed, input.tpr, with the gromacs preprocessor.

```
grompp -f sample.mdp -c merged.gro -p merged.top -o input.tpr
```

Then run

```
g_membed -f input.tpr -p merged.top -xyinit 0.1 -xyend 1.0 -nxy 1000 or
g_membed -f input.tpr -p merged.top -xyinit 0.1 -xyend 1.0 -nxy 1000 -zinit 1.1 -zend 1.0
-nz 100
```

and select the group that has to be inserted. The options of g_membed are `-f` the required input tpr-file, `-p` to update the number of molecules in the topology file, `-xyinit` and `-zinit` the scaling factor for the width and height, respectively, of the protein in the membrane plane at the start of the embedding and `-nxy` and `-nz` the number of steps to reset `-xyinit` and `-zinit` to

`-xyend` and `-zend`, respectively. If the protein is (very) asymmetric in shape use `-ndiff` to remove more lipids from the lower (negative integer) or upper (positive integer) leaflet.

Finally, a 1 ns equilibration run should be run before performing any production runs.

```
grompp -f equi.mdp -c membedded.gro -p merged.top -o equilibrate.tpr
mdrun -deffnm equilibrate
```

When the group to embed is not a default group, such as a protein and its crystal water, an ndx file should also be provided to g_membed. Make sure all the molecule types in the group to embed are unique, e.g., the molecule type of the crystal waters should be different from that of the solvent. Also the freeze and energy exclusion parameters in the mdp file should be changed to match the name of the group to embed.

Note that the program will also issue a warning for most common mistakes we have encountered (will be updated).

References

1. Biggin, P.; Bond, P. Methods Mol Biol 2008, 443, 147.
2. Wallin, E.; von Heijne, G. Protein Sci 1998, 7, 1029.
3. Overington, J.; Al-Lazikani, B.; Hopkins, A. Nat Rev Drug Discov 2006, 5, 993.
4. Carpenter, E.; Beis, K.; Cameron, A.; Iwata, S. Curr Opin Struct Biol 2008, 18, 581.
5. Woolf, T. B.; Roux, B. Proteins 1996, 24, 92.
6. Faraldo-Gómez, J.; Smith, G.; Sansom, M. Eur Biophys J 2002, 31, 217.
7. Shen, L.; Bassolino, D.; Stouch, T. Biophys J 1997, 73, 3.
8. Kandt, C.; Ash, W.; Tieleman, D. Methods 2007, 41, 475.
9. Scott, K.; Bond, P.; Ivetac, A.; Chetwynd, A.; Khalid, S.; Sansom, M. Structure 2008, 16, 621.
10. Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. J Chem Theor Comp 2008, 4, 435.
11. Lau, T.-L.; Kim, C.; Ginsberg, M.; Ulmer, T. EMBO J 2009, 28, 1351.
12. Johansson, M.; Alioth, S.; Hu, K.; Walser, R.; Koebnik, R.; Pervushin, K. Biochemistry 2007, 46, 1128.
13. Deisenhofer, J.; Epp, O.; Sinning, I.; Michel, H. J Mol Biol 1995, 246, 429.
14. Fischer, G.; Kosinska-Eriksson, U.; Aponte-Santamaría, C.; Palmgren, M.; Geijer, C.; Hedfalk, K.; Hohmann, S.; de Groot, B. L.; Neutze, R.; Lindkvist-Petersson, K. PLoS Biol 2009, 7, e1000130.
15. Berger, O.; Edholm, O.; Jühnig, F. Biophys J 1997, 72, 2002.
16. Tieleman, P.; MacCallum, J.; Ash, W.; Kandt, C.; Xu, Z.; Monticelli, L. J Phys: Condens Matter 2006, 18, S1221.
17. Siu, S. W. I.; Vácha, R.; Jungwirth, P.; Böchmann, R. A. J Chem Phys 2008, 128, 125103.
18. Periole, X.; Huber, T.; Marrink, S.-J.; Sakmar, T. P. J Am Chem Soc 2007, 129, 10126.
19. Ollila, O. H. S.; Risselada, H. J.; Louhivuori, M.; Lindahl, E.; Vattulainen, I.; Marrink, S. J Phys Rev Lett 2009, 102, 078101.
20. Kaminski, G.; Friesner, R.; Tirado-Rives, J.; Jorgensen, W. J Phys Chem B 2001, 105, 6474.
21. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Hermans, J. In Intermolecular Forces; Pullman, B., Eds.; D. Reidel Publishing Company: Dordrecht, 1981; pp. 331–342.
22. Sorin, E. J.; Pande, V. S. Biophys J 2005, 88, 2472.
23. Jorgensen, W.; Chandrasekhar, J.; Madura, J.; Impey, R.; Klein, M. J Chem Phys 1983, 79, 926.
24. van Gunsteren, W. F.; Berendsen, H. J. C. Gromos-87 Manual Biomos BV Nijenborgh 4, 1947 AG Groningen, The Netherlands, 1987.
25. Darden, T.; York, D.; Pedersen, L. J Chem Phys 1993, 98, 10089.
26. Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. J Comput Chem 1997, 18, 1463.
27. Bussi, G.; Donadio, D.; Parrinello, M. J Chem Phys 2007, 126, 014101.
28. Berendsen, H.; Postma, J.; DiNola, A.; Haak, J. J Chem Phys 1984, 81, 3684.
29. DeLano, W. The Pymol Molecular Graphics System; DeLano Scientific LLC: Palo Alto, CA, USA, 2008. Available at: http://www.pymol.org.