Supplementary Information: Evaluation of constrained and restrained molecular dynamics simulations methods for predicting skin lipid permeability

Nicola Piasentin [a,b], *Guoping Lian* * [a,b], *Qiong Cai* [a]

[a] Department of Chemical and Process Engineering, University of Surrey, Guildford GU27XH, U.K.

[b] Unilever R&D Colworth, Unilever, Sharnbrook MK441LQ, U.K.

Equilibration of starting system

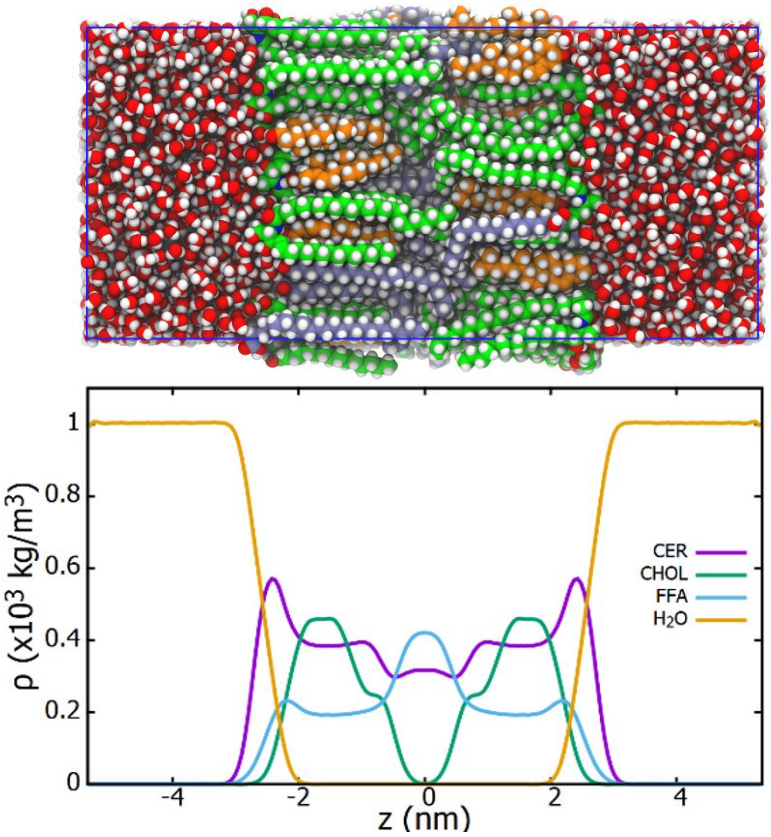

Figure S1 Last frame of the 250 ns unconstrained run and corresponding mass density ρ of the lipid bilayer as a function of the z axis. Molecules are represented via their van der Waals radii. Colour legend: oxygen (red), hydrogen (white), nitrogen (blue), FFA carbon (violet), CHOL

Densities are similar to those published previously in literature [1-2]. Both CER and FFA densities show a peak at the centre of the bilayer, which agrees with the crystalline-liquid state due to tail

interdigitation [3]. Differently, the short and bulky CHOL molecules stay embedded in the leaflets, slightly deeper than the other two lipid families. No CHOL flip-flop events nor lipid molecules leaving the leaflets have been observed during equilibration and production run. The head-to-head thickness of the bilayer d_{HH}, which is defined as the distance among the density peaks of the nitrogen atoms in CER heads, is $d_{HH} = 49.0 \pm 0.4$ Å, which agrees with the previously published value of $d_{HH} = 49.1 \pm 0.1$ Å [4]. The average area per lipid is $A_{AVE} = 32.5 \pm 0.2$ Å^2 and the average CHOL tilting with respect to the z axis is $\theta_{CHOL} = 8.6^\circ \pm 0.9^\circ$, which are in good agreement with the values reported in literature of $A_{AVE} = 32.81 \pm 0.01$ Å^2 and $\theta_{CHOL} = 8.9^\circ \pm 0.1^\circ$, respectively [4].

Partition coefficient calculation

The simulated system is divided into n slices and the contribution of each volume element to the total lipid/water partition coefficient $K_{lip/wat}$ is calculated by considering the individual contribution $K^i_{lip/wat}$ of the i-th slice which is proportional to the local free energy difference following the relationship

$$K^{(i)}_{lip/wat} \propto \exp\left[-\frac{\Delta G_i}{RT}\right] \quad \text{(S1)}$$

In this work, the product RT is used rather than $\beta^{-1} = k_B T$ since the natural units for energy in MD simulations are in kJ mol^{-1}.

Thus, $K_{lip/wat}$ can be expressed as [5]

$$K_{lip/wat} = \frac{\sum_{j=1}^{n_{wat}} V(z_j)}{\sum_{i=1}^{n_{lip}} V(z_i)} \times \frac{\sum_{i=1}^{n_{lip}} V(z_i) \exp\left[-\frac{\Delta G_i}{RT}\right]}{\sum_{j=1}^{n_{wat}} V(z_j) \exp\left[-\frac{\Delta G_j}{RT}\right]} \quad \text{(S2)}$$

Where $V(z_i)$ is the volume of the i-th slice, n_{lip} is the total number of slices containing the lipid phase and n_{wat} is the total number of slices containing the water phase, with $n = n_{lip} + n_{wat}$.

In this work, the total number of windows explored via constrained/restrained MD is n = 90. Since they are equispaced along z, with distance $\delta z_k = \delta z = 0.1$ nm $\forall$ k = 1, …, 90 and they have a square section with side of length l, the individual volumes are independent of the slice, that is

$$V(z_k) = \delta z_k \times l^2 = \delta z \times l^2 = \delta V \quad \text{(S3)}$$

This leads to

$$K_{\frac{lip}{wat}} = \frac{\frac{n_{wat} \times \delta V}{\delta V \times \sum_{i=1}^{n_{lip}} \exp\left[-\frac{\Delta G_i}{RT}\right]}}{\frac{n_{lip} \times \delta V}{\delta V \times \sum_{j=1}^{n_{wat}} \exp\left[-\frac{\Delta G_j}{RT}\right]}} = \frac{n_{wat}}{n_{lip}} \frac{\sum_{i=0}^{n_{lip}} \exp\left[-\frac{1}{RT}\Delta G(z_i)\right]}{\sum_{j=0}^{n_{wat}} \exp\left[-\frac{1}{RT}\Delta G(z_j)\right]} \quad \text{(S4)}$$

Which is the reported equation. The phase separation is set by $|\Delta G| > 0.2$ kJ mol^{-1}, where lower values are counted as water phase and higher ones as lipid phase [5].

Eq. (S2) from ref. [5] represents the averaged partition across the membrane. If one considers that the water phase slices have $\Delta G \approx 0$, then $\exp[-(\Delta G_j)/RT] \approx 1$ and therefore Eq. (S2) reduces to the following

$$K_{lip/wat} = \frac{\sum_{j=1}^{n_{wat}} V(z_j)}{\sum_{i=1}^{n_{lip}} V(z_i)} \times \frac{\sum_{i=1}^{n_{lip}} V(z_i) \exp\left[-\frac{\Delta G_i}{RT}\right]}{\sum_{j=1}^{n_{wat}} V(z_j)} \quad \text{(S5)}$$

Which simplifies to

$$K_{lip/wat} = \frac{1}{\sum_{i=1}^{n_{lip}} \delta z} \sum_{i=1}^{n_{lip}} \delta z \exp\left[-\frac{\Delta G_i}{RT}\right] = \frac{1}{L} \sum_{i=1}^{n_{lip}} \delta z \times K_{lip/wat}^{(i)} \approx \frac{1}{L} \int_0^L K(z)\, dz \quad \text{(S6)}$$

Where L is the bilayer length, which is the same averaged partition coefficient reported in [6].

Perpendicular diffusion coefficient profiles D_z

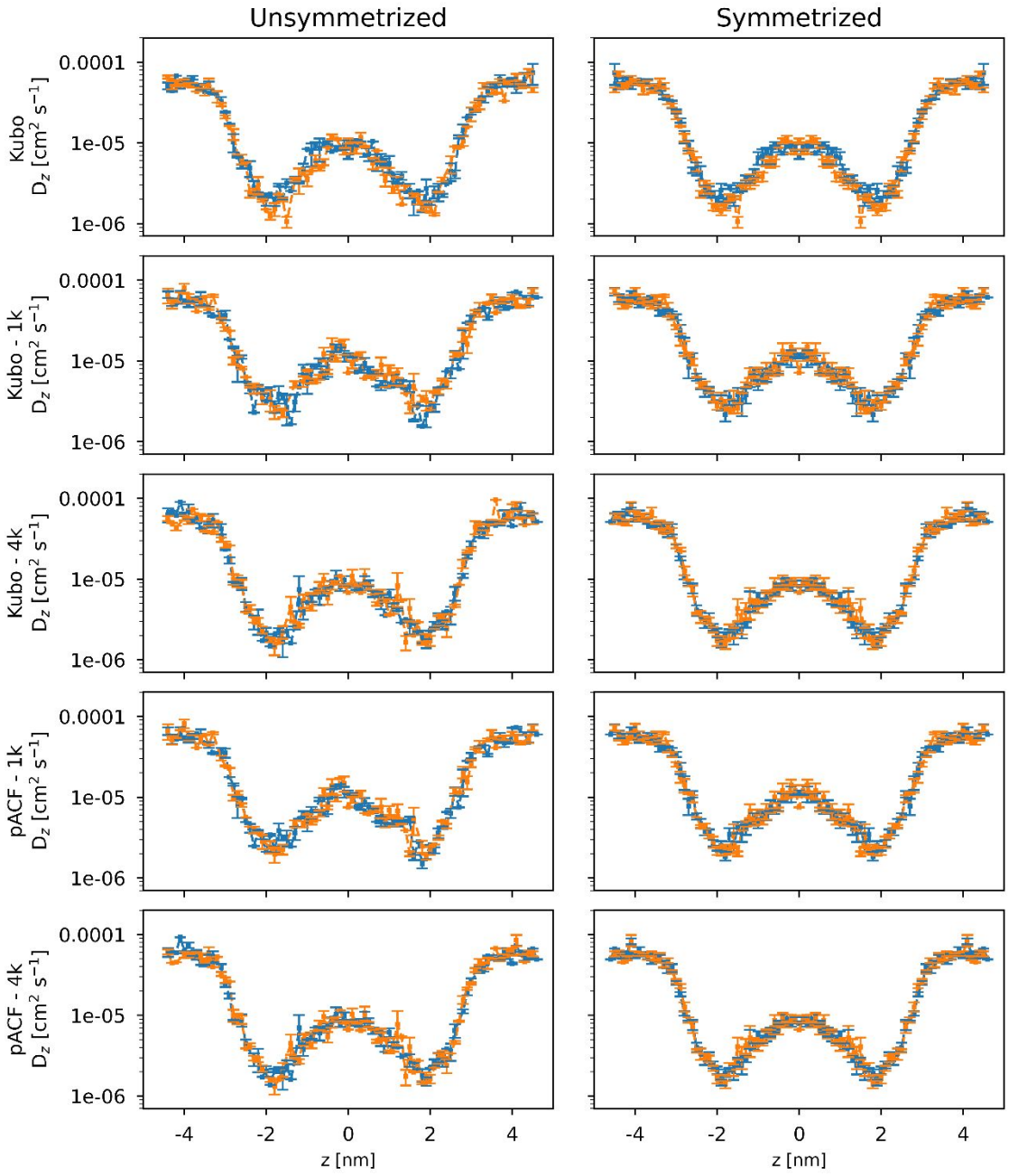

Figure S2 Unsymmetrised (left) and symmetrised (right) perpendicular diffusion coefficient profiles for all the methods tested and all the data sets (I data set - blue, II data set - orange).

Average force profiles

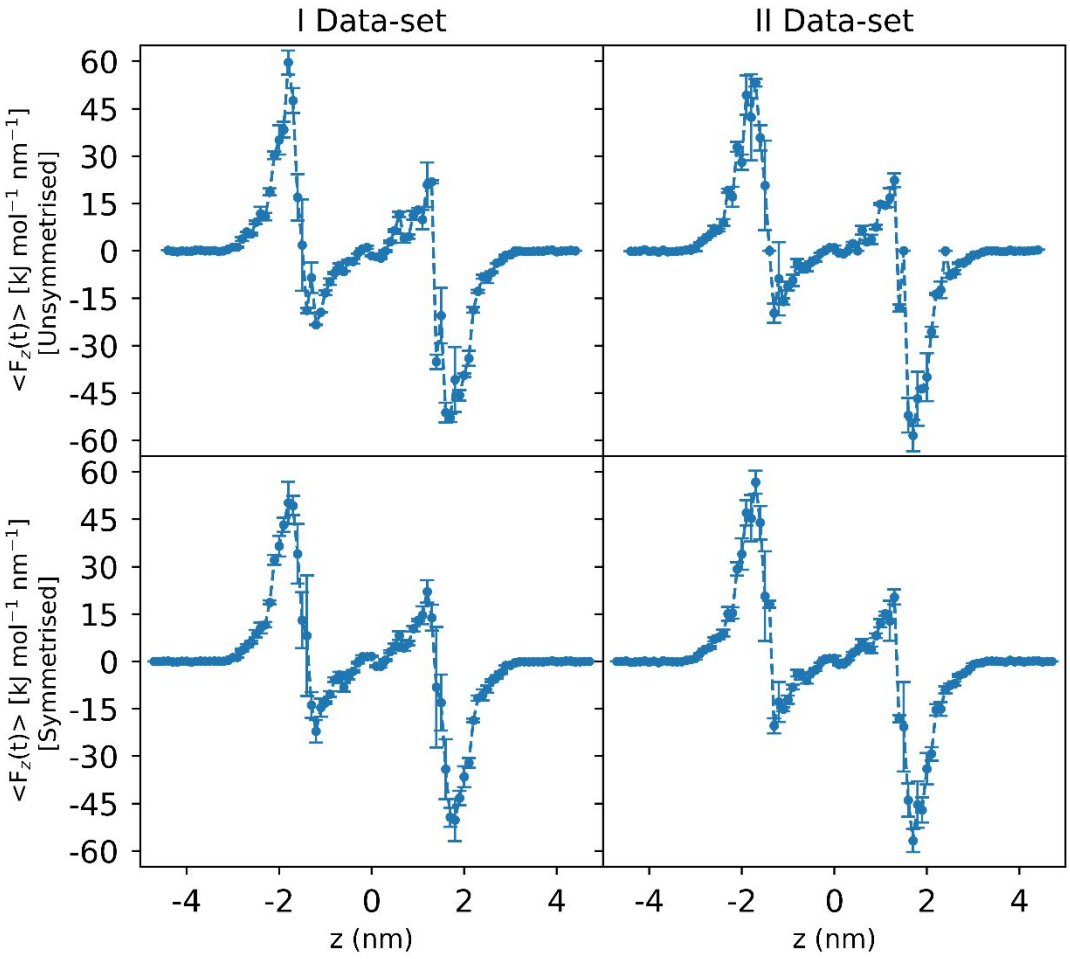

Figure S3 Unsymmetrized and symmetrized average of forces with respect to *z* profiles for PMcF method and all the data sets.

Potential of the mean force profiles for PMcF and WHAM

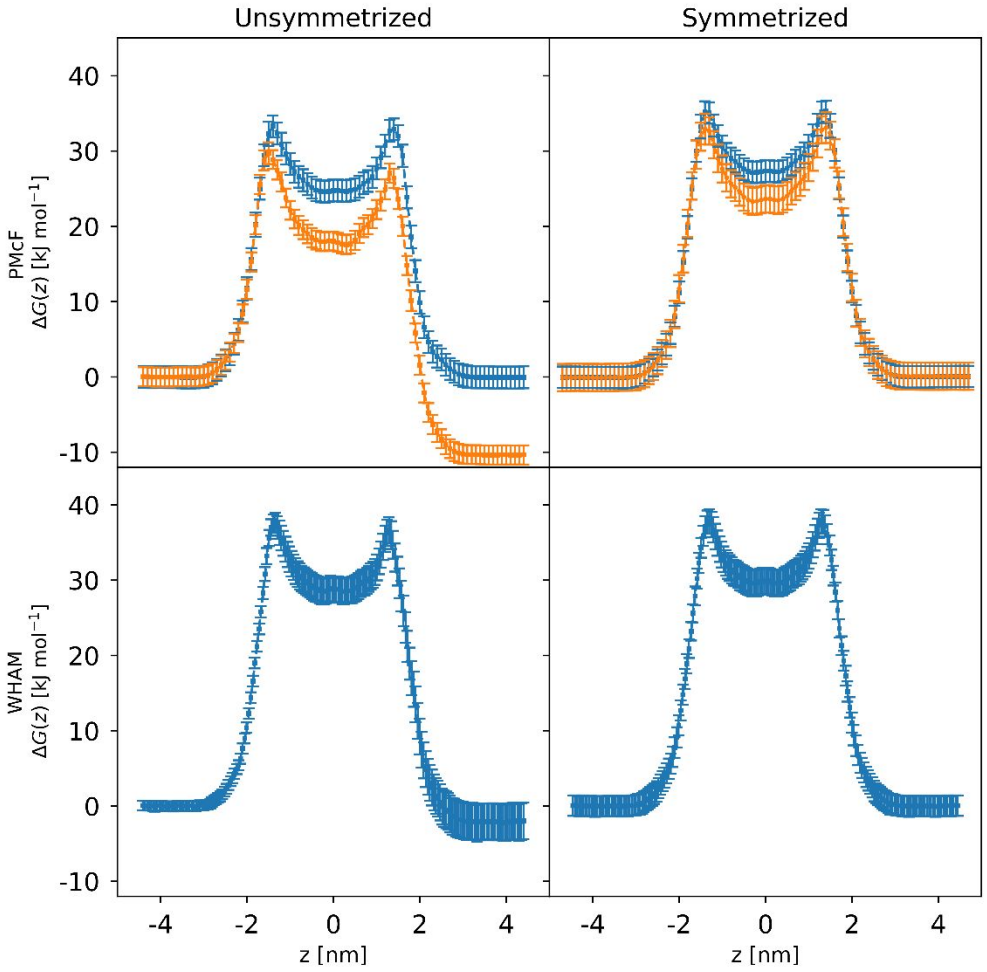

Figure S4 Unsymmetrized and symmetrized ΔG profiles for all the methods tested and all the data sets (I data set - Orange, II data set - Blue, WHAM has only one data set).

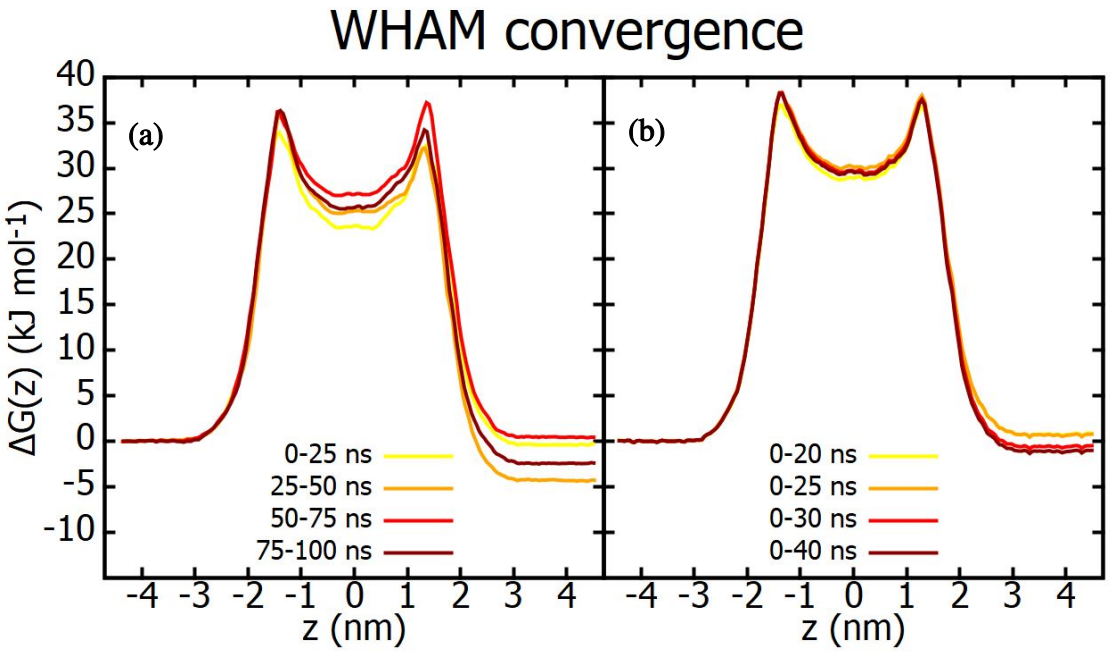

Figure S5 Check for WHAM convergence. (a) First 100 ns of equilibration. Once sub-runs of 25ns differ from less than $1k_BT$ the equilibration is considered as achieved. The production starts. (b) Following 40 ns of production. Adding further data changes the profile by $<< 1k_BT$, therefore the sampling is considered completed and the profile computed and symmetrised.

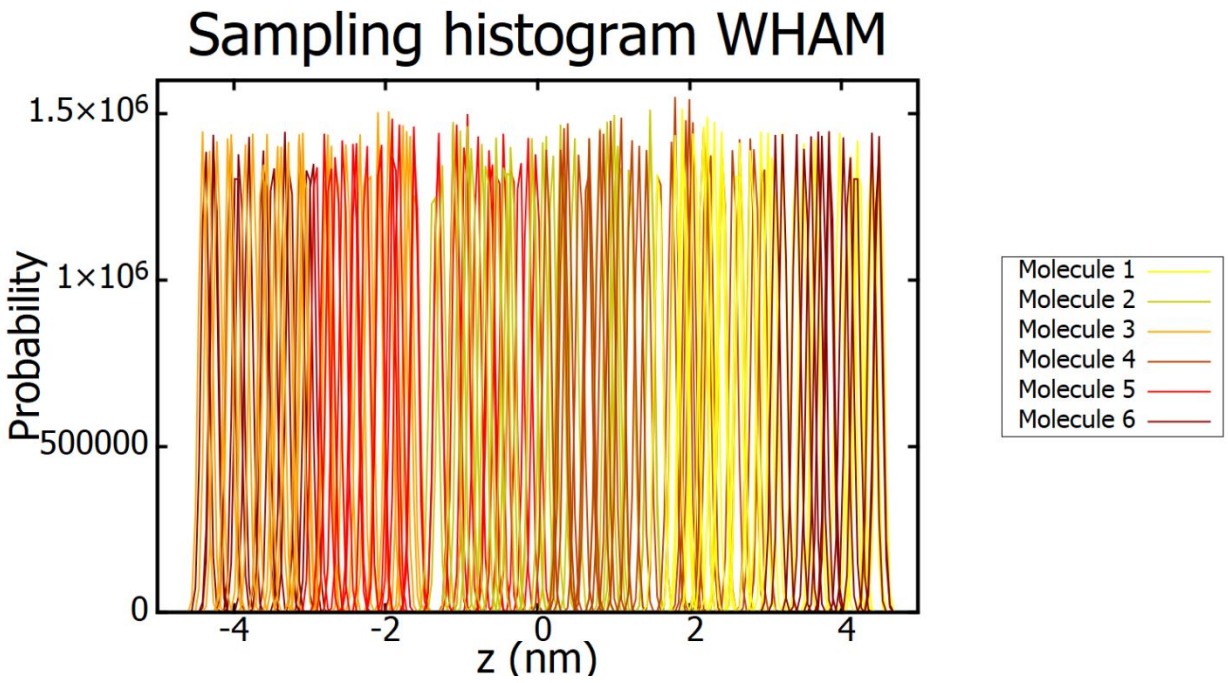

Figure S6 Check for WHAM completeness. All probability histogram for all the molecules from all the windows. The reaction coordinate is completely covered, i.e., the free energy profile along the RC is sampled in its length and no regions are left uncovered.

Permeability data from literature and temperature correction

In Table S1 are collected values of permeability P^o from previous computational data. To draw a meaningful comparison, the data is corrected for the temperature difference as [7-8]

$$\log P_{303K} = \log P_T + 0.04 \times (303 - T) \quad \text{(S7)}$$

Where P_T is the permeability at temperature T and P_{303K} is the permeability at T = 303 K. Temperature-corrected permeabilities P are reported in Table S1, where all original literature permeability coefficients P^o are shifted to T = 303 K. For the large temperature shift of $\Delta T = 50$ K required in Das *et al.* work [9] the applicability of Eq. (S7) is not necessarily verified. We can partially verify its applicability by applying Eq. (S7) to the results for the 2:2:1 bilayer at T = 350 K in the same work, which gives a corrected permeability of P = 1.7 x 10^{-9} cm s^{-1}. From the same work, the temperature-corrected value for the 2:2:1 at T = 300 K is P = 4.9 x 10^{-9} cm s^{-1}. Therefore Eq. (S7) predicts a correction factor of ca. 1/76, while the ratio of the two values from the simulations predicts a correction factor of ca. 1/27. Although the order of magnitude is the same, we decided to use the simulation ratio correction of 1/27 to shift results from 350 K to 300 K in [9], because of its natural link to MD results. For this reason, in Table S1 the results for 2:2:1 at T = 350 K corrected for the temperature is missing (-).

When possible, the reported ΔG and D_z profiles have been digitized to recalculate the permeability coefficients reported in the published works. For Das *et al.* [9], not all the profiles are reported. However, from the ones that was possible to digitize we found a discrepancy of 4 orders of magnitude from the reported profiles with respect to the reported permeability values, as also implicitly noted by Lundborg *et al.* [10]. Therefore, we shifted by 4 orders of magnitude all the results collected from this work. Both results from Gupta *et al.* [2, 11] agree with the recalculation, therefore no correction was needed, and no data is reported in Table S1 (-). Recalculation of profiles reported in [1] gives a difference of 7 orders of magnitude that we applied to the original value after correcting for the temperature difference. No profiles are reported by Lundborg *et al.* [10], however the profiles were sent to us by personal communication and confirm the value reported in the original paper, therefore no corrected value is reported (-).

Table S1 Original permeability coefficients P^o from MD literature, corresponding temperature-corrected permeability coefficients P and unit of measurement corrected P^c for a water molecule across a hairpin SC lipid bilayer[1].

Model	T [K]	P^o [cm s^{-1}]	P [cm s^{-1}]	P^c [cm s^{-1}]	Method	Forcefield	Ref.
1:0:0	350	1.1×10^{-8}	5.4×10^{-10}	5.4×10^{-6}	C	GR87+BR	9
1:0:0	310	1.12×10^{-6}	5.9×10^{-7}	-	C	GR87+BR	11
1:1:1*	303.15	1.1×10^{-5}	1.1×10^{-5}	-	S	CHARMM36	10
1:1:1	350	8.2×10^{-8}	4.0×10^{-9}	4.0×10^{-5}	C	GR87+BR	9
1:1:1	305	3.3×10^{-10}	2.7×10^{-10}	3.0×10^{-3}	C	GR87+BR	1
1:1:1	310	7.08×10^{-5}	3.7×10^{-5}	-	C	GR54a6/a7+BR	2
2:2:1	350	1.3×10^{-7}	-	1.3×10^{-3}	C	GR87+BR	9
2:2:1	300	3.7×10^{-9}	4.9×10^{-9}	4.9×10^{-5}	C	GR87+BR	9

[1] All corrected P values are reported for T = 303 K by applying a temperature correction as in [7-8] if needed, exception made for T = 350 K (see this section). Missing terms (-) are considered in the next session. Model indicates system's composition in terms of molar ratio CER:CHOL:FFA, with CER [NS] 24:0/18:1, cholesterol and protonated lignoceric acid. Method indicate the MD method used, constrained MD (C, PMcF + Kubo [12-14]) and steered MD (S, forward-reverse method [8, 10, 15-16]). Forcefield abbreviations are: GROMOS-87 (GR87, [17]), GROMOS-54a6/a7 (GR54a6/a7, [18]), CHARMM36 [19] and the Berger set for lipids (BR, [20]) which also includes the parameters set for free fatty acids and cholesterol from [21].

* Ceramide is CER [NP] 24:0/18:0.

Python script README

This is a brief introduction to how the script works and what can be calculated with it. The script has been developed in python 3.6.9 on Ubuntu 18.04 LTS and requires the two external packages numpy (v. 1.19.5) and tqdm (v. 4.55.1), plus the standard module sys, argparse, and subprocess. The script makes calls to GROMACS functions, therefore a functioning installation of GROMACS is required. The code was developed with GROMACS version 2018.1, but in principle it should work with any GROMACS version after (and including) 5.0.

The tool is based on GROMACS output. One or more molecules need to be restrained or constrained in a set of simulations whose outputs need to be listed in text files and given as input to the script. It is implied that these simulations must share the same temperature, length, spring force constant, type of constrain, molecule type of permeant etc., otherwise the code will fail and/or produce unreliable outcome.

Standard input that can be required is: *.tpr binary files, *_pullx.xvg files, *_pullf.xvg files, *.xtc position files and an .ndx file in GROMACS index format. The files must be in the same order in all the file lists, but the script doesn't check for this consistency. The mechanism behind the usage is like those of other GROMACS tools, e.g., gmx wham or gmx bar. Output from the

script will be saved in the directory ‘script_output’ which will be made in the same directory from where the script is being run. The script generates directories and save files. The user must therefore have the needed permits to run the analysis.

A quick explanation of what the code does, what are the input files required and which options are available can be printed on screen with ‘--h', e.g., *$> python3 script.py --h*.

The options are reported in the following table, and then a brief description of their input/ output and workflow is summarised.

Option	Description
-h	show this help message and exit
-it	List of .tpr files in order.
-ix	List of pullx.xvg files in order.
-if	List of pullf.xvg files in order.
-diff_perp_kubo	Specify to calculate the perpendicular diffusion profile with input data with the Kubo relation.
-pmf	Specify to calculate ΔG with input data via potential of the mean constraint force.
-fACFlen	Step length of force autocorrelation function integration. Default is 10000.
-avelen	Step length of force average time calculation. Default is 1000.

-flen	Amount of data used to calculate average for $\langle F(z) \rangle_t$ values in PMF calculation. Default is 0.50 (half data).
-bw	Bin width (nm) for binning profiles. Default is 0.1 nm.
-com	Shift of the COM (nm) for symmetrisation of profiles. Default is 0 nm.
-dG	Shift of the PMF (kJ/mol). Default is 0 kJ/mol.
-diff_lat	Specify to calculate the lateral diffusion profile via the MSD relation.
-xtc	List of .xtc files in order.
-n	Index file in GROMACS format.
-natoms	Number of atoms per pull group.
-lateral	Direction perpendicular to plane of diffusion. Default is z.
-nave	Number of subruns to subdivide the mean squared displacement calculation.
-e	Amount of data used to calculate MSD. Default is the whole dataset.
-dllen	Amount of data used to calculate average for lateral diffusion coefficients. Default is 0.50.
-gmx	GROMACS binary to run the commands.
-diff_perp_pACF	Specify to calculate the perpendicular diffusion profile with input data via pACF.
-pACF_len	Step length of positions autocorrelation function integration. Default is 10000.

GROMACS binaries

Different tools from GROMACS library are utilized, therefore it is expected that GROMACS is installed, and the corresponding binaries are sourced. The standard command used by this script is 'gmx', e.g., ‘gmx analyze’. Please check if the binaries are correctly sourced by calling 'gmx --h'. If not, source them before running this script or give the full path by specifying the option *-gmx* (e.g. *-gmx /usr/bin/GMRX*). If GROMACS was compiled with PMI, e.g., in a computational cluster, specify *-gmx gmx_mpi* and source the binary or specify the full path.

Perpendicular diffusion coefficient profile via Kubo relation

To calculate the perpendicular diffusion profile via Kubo relation, specify the option *-diff_perp_kubo*. The required inputs are the following lists of files: tpr (*-it*), pullx (*-ix*) and pullf (*-if*). The diffusion coefficients are calculated via the autocorrelation function of the force acting on the centre of mass of the molecules [14]. The code takes the input parameters in the tpr files and checks their consistency. Then it loads the pullx files and computes the average position of the molecules, that is the window positions for the profile. Finally, it loads the pullf files and computes the ACFs of the force acting on the centre of mass of the molecules. ACFs are computed via GROMACS' gmx analyze tool and the results stored. The force ACFs are then loaded and integrated with the trapezium rule and the individual diffusion values per window are saved. The

values are collected into a profile, that is saved both as is and symmetrized with respect to the 0 of the window positions.

Errors are calculated by taking average and deviation if more than one value for the diffusion profile falls in one window. In case only one value is present in a window, no error estimate is given.

Force ACFs are integrated along the first 10000 points, i.e., a time length of integration equal to 10000 times the time step of integration of the simulation and the output timestep. The number of points can be changed by specifying *-fACFlen*. The individual calculated values are binned to obtain a global profile. The bin width is 0.1 nm by default. It should roughly correspond to the original distance of the windows. It can be changed by specifying *-bw* and a value in nm. The symmetry point for the profiles is 0 by default, it can be changed by specifying *-com* and a value in nm. When saving the force ACFs, the code checks the ratio among the maximum values of the force ACFs at the beginning and at the end of the integration time. The values of the ratios are store in the corresponding output files. If the ratio is more than 5%, the code warns the user about a slow convergence in the fACFs. In this case, please do consider longer integration times.

Output from gmx analyze is saved as

./script_output /fACF_gmx_analyze_tlength_#ps /facf_#sim_tlength_#steps.xvg

On screen output coming from GROMACS is saved in the same directory as

./gmx_fACF_analyze_output.txt

Moreover the following files are saved in *./script_output*:

- *./Dperp_kubo_individual_molecule_#mol.txt* Contains the individual perpendicular diffusion coefficients per molecule calculated via Kubo relation.
- *./fACF_molecule_#mol.txt* Contains all the fACF for every window for the given molecule as a function of time. Also contains the average window position and the rate of convergence of the fACF.
- *./Dperp_kubo_unsym.txt* Unsymmetrized profile obtained by collecting all the results. Three columns: window position, value and error (1 standard deviation). If more than one value per bin is found, the arithmetic average is returned. Error is calculated as the standard deviation of the mean. No error is estimated if only one value is present.
- *./Dperp_kubo_sym.txt* Same as the un-symmetrized version, but the profile is computed and symmetrized.

Perpendicular diffusion coefficient profile with position-autocorrelation function method

To calculate the perpendicular diffusion profile via pACF, specify the option *-diff_perp_pACF*. The required inputs are the following lists of files: tpr (*-it*) and pullx (*-ix*). The diffusion coefficients are calculated via the autocorrelation function of the position of the centre of mass of the molecules [22-24]. The code takes the input parameters in the tpr files and checks their consistency. Then it loads the pullx files and computes the average position of the molecules, that is the window positions for the profile. Then it computes the variance of the positions and the position autocorrelation functions of the coordinate of the centre of mass of the molecules. Position ACFs are computed via GROMACS' gmx analyze tool and the results stored. The position ACFs are then loaded and integrated with the trapezium rule and the individual diffusion values per window are saved. The values are collected into a profile, that is saved both as is and symmetrized with respect to the 0 of the window positions.

Errors are calculated by taking average and deviation if more than one value for the diffusion profile falls in one window. In case only one value is present in a window, no error estimate is given.

Position ACFs are integrated along the first 10000 points, i.e., a time length of integration equal to 10000 times the time step of integration of the simulation and the output timestep. The number

of points can be changed by specifying *-pACFlen*. The individual calculated values are binned to obtain a global profile. The bin width is 0.1 nm by default. It should roughly correspond to the original distance of the windows. It can be changed by specifying *-bw* and a value in nm. The symmetry point for the profiles is 0 by default, it can be changed by specifying *-com* and a value in nm. When saving the position ACFs, the code checks the ratio among the maximum values of the pACFs at the beginning and at the end of the integration time. The values of the ratios are store in the corresponding output files. If the ratio is more than 5%, the code warns the uses about a slow convergence in the pACFs. In this case, please do consider longer integration times.

Output from gmx analyze is saved as

./script_output /pACF_gmx_analyze_tlength_#ps /acf_#sim_tlength_#steps.xvg

On screen output coming from GROMACS is saved in the same directory as

gmx_pACF_analyze_output.txt

Moreover the following files are saved in *./script_output*:

- *./Dperp_pACF_individual_molecule_#mol.txt* Contains the individual perpendicular diffusion coefficients per molecule calculated via pACF method.

- *./pACF_molecule_#mol.txt* Contains all the pACFs for every window for the given molecule as a function of time. Also contains the average window position and the rate of convergence of the pACF.
- *./Dperp_pACF_unsym.txt* Unsymmetrized profile obtained by collecting all the results. Three columns: window position, value and error (1 standard deviation). If more than one value per bin is found, the arithmetic average is returned. Error is calculated as the standard deviation of the mean. No error is estimated if only one value is present.
- *./Dperp_pACF_sym.txt* Same as the un-symmetrized version, but the profile is computed and symmetrized.

ΔG via potential of the mean constraint force

To calculate the potential of the mean force, specify the option *-pmf*. The required inputs are the following lists of files: tpr (*-it*), pullx (*-ix*) and pullf (*-if*). The potential of the mean force profile is calculated by using the average of the force acting on the centre of mass of the molecules [13]. The code takes the input parameters in the tpr files and checks their consistency. Then it loads the pullx files and computes the average position of the molecules, that is the window positions for the profile. Finally, it loads the pullf files and computes the average of the force <F(z)> acting on the

centre of mass of the molecules. The individual <F(z)> vs time curves for the molecules are computed and saved. The final values for <F(z)> are derived by averaging on the last part of the <F(z)> vs time plots, where it is expected that the average of the force has reached a steady value. The values are collected into a force profile, that is saved both as is and symmetrized with respect to the 0 of the window positions. The potential of the mean force is calculated by integrating the force profile. The calculations are carried out and saved both for the symmetrized and unsymmetrized force inputs. Errors are calculated by propagating the error on the integral carried on from the left and the right. The final output files for the symmetric/unsymmetric profiles contain the following: window position, potential of the mean force integrated from left and associated error, potential of the mean force calculated from right and associated error and average with propagated error.

The <F(z)> vs time plots are calculated with the input force values divided in 1000 blocks, i.e., the output are 1000 couples of (time, <F(z)>(t)) points. The length can be changed by specifying -*avelen* and giving a positive integer. Similarly, the final value of <F(z)> is calculated on the second half of the (time, <F(z)>(t)) curve. It can be changed by specifying -*flen* and the normalized amount of data that is wanted, e.g., 0.60 to use 60% of the curve for <F(z)> calculation. The

individual calculated force values are binned to obtain a global profile. The bin width is 0.1 nm by default. It should roughly correspond to the original distance of the windows. It can be changed by specifying *-bw* and a value in nm. The symmetry point for the profiles is 0 by default, it can be changed by specifying *-com* and a value in nm. The potential of the mean force profile is computed as is and is not shifted vertically. This can be done by specifying *-dG* and an energy value in kJ mol^{-1}. Please note that the shift is eventually going to change the resistivity/permeability values, if calculated.

Moreover the following files are saved in *./script_output*:

- *./force_average_molecule_#mol.txt* Contains the individual force averages <F(z)> per molecule with the associated error calculated as the standard deviation of the average force vs time plot.
- *./force_time_molecule_#mol.txt* Contains all the time averages for every window for the given molecule as a function of time. Also contains the average window position.
- *./force_unsym.txt* Unsymmetrized profile obtained by collecting all the results. Three columns: window position, value and error (1 standard deviation). If more than one value per bin is found, the arithmetic average is returned. Error is calculated as the standard

deviation of the mean. If only one value is present, the error is calculated as the standard deviation of the force average vs time curve.

- *./force_sym.txt* Same as the un-symmetrized version, but the profiles is computed and symmetrized.
- *./PMF_unsym_control.txt* Potential of the mean force obtained by integrating the unsymmetrized profile of the average of forces, i.e. via potential of the mean constraint force. There are 7 columns: window position, ΔG as integrated from the left (negative z values), associated error, ΔG as integrated from the right (positive z values), associated error, resulting ΔG obtained by averaging the two directions of integration and associated error. Not so useful per se, but if the profiles are not all the same then probably something went wrong in the data or the script. Errors are calculated from standard error propagation by propagating the variance of the averages through the trapezium rule integration.
- *./PMF_sym_control.txt* Same as the unsymmetrized version but starting from the integration of symmetric force average curve.
- *./PMF_unsym.txt* Potential of the mean force obtained by integrating the unsymmetrized profile of the average of forces. There are 3 values: window position, ΔG as integrated

from the right (negative z values) and associated error. It is basically the *PMF_unsym_control.txt* file with just the final needed values.

- *./PMF_sym.txt* Same as the unsymmetrized version but starting from the integration of symmetric force average curve.

Resistivity and permeability

If both the perpendicular diffusion coefficient and potential of the mean force profiles calculations are specified with *-diff_perp_kubo* and/or *-diff_perp_pACF* and *-pmf*, the script calculates the resistivity and the permeability via the inhomogeneous solubility diffusion model [14]. In case the perpendicular diffusion profile with both Kubo and pACF methods are requested, the code evaluates resistivity and permeability with both diffusion profiles. Only symmetric profiles are used in this case to avoid missing terms in the integration due to windows not sampled. Results are printed on screen and saved in the summary file *./script_output/analysis_details.txt*. Errors are calculated via standard error propagation, i.e., by propagating variances of free energy and diffusion profiles.

Lateral diffusion coefficient profile via MSD

To calculate the lateral diffusion profile, specify the option *-diff_lat*. The required inputs are the following lists of files: tpr (*-it*), pullx (*-ix*) and xtc (*-xtc*). Moreover, the following parameters are needed: the number of atoms per group (*-natoms*) and a special GROMACS index file (*-n*). The number of atoms per group is the number of atoms of the pull groups of interest. For example, in an all-atoms simulation where one or more water molecules are constrained/restrained at a given distance from a lipid bilayer, you should specify *-natoms* 3 (two hydrogens and one oxygen). The index file given as an input should be built by the user via gmx make_ndx and must contain only one group with all the atoms of the pulled molecules. For example, in an all-atoms simulation where five water molecules are constrained the group must contain the 15 (5x3) indexes of all the water atoms. The diffusion coefficients are calculated via the Einstein formula as the time limit of the mean squared displacement of the centre of mass of the molecules. The code takes the input parameters in the tpr files and checks their consistency. Then it loads the pullx files and computes the average position of the molecules, that is the window positions for the profile. Then, the script reduces the tpr and xtc files via gmx convert-tpr and gmx trjconv commands and extracts the coordinates of the molecules of interest from the xtc files via gmx traj. The mean squared displacement of the molecules is then computed from the coordinate files and the corresponding

time limits are calculated and saved per individual molecule. The lateral diffusion coefficients are derived by averaging on the final part of the mean squared displacement time limit plot and are then saved individually per molecule. The values are collected into a profile, that is saved both as is and symmetrized with respect to the 0 of the window positions. Errors are calculated by propagating the error (variance) of the averages over to the final profile.

The individual calculated values are binned to obtain a global profile. The bin width is 0.1 nm by default. It should roughly correspond to the original distance of the windows. It can be changed by specifying *-bw* and a value in nm. The symmetry point for the profiles is 0 by default, it can be changed by specifying *-com* and a value in nm. The script assumes that the plane of interest for the lateral diffusion calculation is perpendicular to the z axis. To choose another axis, specify *-lateral (x,y,z)* (default is *-lateral z*). To enhance statistics, it is possible to split a single run for a molecule into N sub runs. The default is to consider just one molecule, to change this specify *-nave* followed by the number of virtual molecules to be considered. If *-nave* is greater than 1, it is better to separate sequential calculations to decorrelate the values. By default, the code considers all the data. To change this, specify *-e* and a value in between 0 (discard all data) and 1 (consider all data). To calculate the individual lateral diffusion coefficients, an average of the mean squared

displacement vs time plot is taken. Since the starting values are usually noisy, the first 50% (i.e., 0.5) of the plot is not considered. To change this, specify *-dllen* and a value in between 0 (average on whole data) and 1 (discard all).

Output from the analysis is saved in the subdirectory *./script_output/gmx_lateral_diffusion*. The following files are also saved in this subdirectory:

- *./red_#win.tpr* Output coming from gmx convert-tpr, that is the reduced tpr for the #window.
- *./red_#win.xtc* Output coming from gmx trjconv, that is the reduced xtc for all the restrained/constrained molecules in the #window.
- *./original_index.ndx* A local copy of the original index supplied via *-ndx*.
- *./local_index.ndx* A local index generated by the script that is used to extract all the coordinates.
- *./coord_file_#window_#mol.xvg* Output coming from gmx traj, that is the xvg GROMACS output file with the coordinate of the centre of mass of the molecule #mol in the window #window as a function of time. The value related to the orthogonal degree

of freedom (e.g., the z value in one is interested in the diffusion along the plane orthogonal to z) is not plotted.

Moreover the following files are saved in *./script_output*:

- *./Dlat_individual_molecule_#mol.txt* Contains the individual lateral diffusion coefficients per molecule calculated via Einstein method. Also contains the associated error calculated as the standard deviation of the mean squared displacement vs time plot.
- *./MSD_molecule_#mol.txt* Contains all the calculated mean squared displacements for every window for the given molecule as a function of time. Also contains the average window position.
- *./Dlat_unsym.txt* Unsymmetrized profile obtained by collecting all the results. Three columns: window position, value and error (1 standard deviation). If more than one value per bin is found, the arithmetic average is returned. Error is calculated as the standard deviation of the mean. If only one value is present, the error is calculated as the standard deviation of the force mean squared displacement vs time curve.
- *./Dlat_sym.txt* Same as the un-symmetrized version, but the profile is computed and symmetrized.

References

1. Del Regno, A.; Notman, R., Permeation pathways through lateral domains in model membranes of skin lipids. *Phys Chem Chem Phys* **2018,** *20* (4), 2162-2174.
2. Gupta, R.; Sridhar, D. B.; Rai, B., Molecular Dynamics Simulation Study of Permeation of Molecules through Skin Lipid Bilayer. *J Phys Chem B* **2016,** *120* (34), 8987-96.
3. Das, C.; Noro, M. G.; Olmsted, P. D., Simulation studies of stratum corneum lipid mixtures. *Biophys J* **2009,** *97* (7), 1941-51.
4. Wang, E.; Klauda, J. B., Simulations of Pure Ceramide and Ternary Lipid Mixtures as Simple Interior Stratum Corneum Models. *J Phys Chem B* **2018,** *122* (10), 2757-2768.
5. Jakobtorweihen, S.; Ingram, T.; Smirnova, I., Combination of COSMOmic and molecular dynamics simulations for the calculation of membrane–water partition coefficients. *Journal of computational chemistry* **2013,** *34* (15), 1332-1340.
6. Mitragotri, S., A theoretical analysis of permeation of small hydrophobic solutes across the stratum corneum based on Scaled Particle Theory. *J Pharm Sci* **2002,** *91* (3), 744-52.
7. Abraham, M. H.; Martins, F., Human skin permeation and partition: general linear free-energy relationship analyses. *J Pharm Sci* **2004,** *93* (6), 1508-23.
8. Lundborg, M.; Wennberg, C. L.; Narangifard, A.; Lindahl, E.; Norlen, L., Predicting drug permeability through skin using molecular dynamics simulation. *J Control Release* **2018,** *283*, 269-279.
9. Das, C.; Olmsted, P. D.; Noro, M. G., Water permeation through stratum corneum lipid bilayers from atomistic simulations. *Soft Matter* **2009,** *5* (22), 4549-4555.
10. Lundborg, M.; Narangifard, A.; Wennberg, C. L.; Lindahl, E.; Daneholt, B.; Norlen, L., Human skin barrier structure and function analyzed by cryo-EM and molecular dynamics simulation. *J Struct Biol* **2018,** *203* (2), 149-161.
11. Gupta, R.; Dwadasi, B. S.; Rai, B., Molecular Dynamics Simulation of Skin Lipids: Effect of Ceramide Chain Lengths on Bilayer Properties. *J Phys Chem B* **2016,** *120* (49), 12536-12546.

12. Kubo, R., The fluctuation-dissipation theorem. *Reports on progress in physics* **1966,** *29* (1), 255.

13. Berendsen, H. J. C.; Marrink, S. J., Molecular dynamics of water transport through membranes: Water from solvent to solute. *Pure and Applied Chemistry* **1993,** *65* (12), 2513-2520.

14. Marrink, S. J.; Berendsen, H. J., Simulation of water transport through a lipid membrane. *The Journal of Physical Chemistry* **1994,** *98* (15), 4155-4168.

15. Kosztin, I.; Barz, B.; Janosi, L., Calculating potentials of mean force and diffusion coefficients from nonequilibrium processes without Jarzynski's equality. *J Chem Phys* **2006,** *124* (6), 64106.

16. Chen, L. Y., Nonequilibrium fluctuation-dissipation theorem of Brownian dynamics. *J Chem Phys* **2008,** *129* (14), 144113.

17. Van Gunsteren, W. F.; Berendsen, H. J. C., Gromos-87 manual. *BIOMOS bv Nijenborgh* **1987,** *4*, 9747.

18. Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; van Gunsteren, W. F., Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur Biophys J* **2011,** *40* (7), 843-56.

19. Huang, J.; MacKerell, A. D., Jr., CHARMM36 all-atom additive protein force field: validation based on comparison to NMR data. *J. Comput. Chem.* **2013,** *34* (25), 2135-45.

20. Berger, O.; Edholm, O.; Jähnig, F., Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature. *Biophysical Journal* **1997,** *72* (5), 2002-2013.

21. Höltje, M.; Förster, T.; Brandt, B.; Engels, T.; von Rybinski, W.; Höltje, H.-D., Molecular dynamics simulations of stratum corneum lipid models: fatty acids and cholesterol. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **2001,** *1511* (1), 156-167.

22. Woolf, T. B.; Roux, B., Conformational Flexibility of o-Phosphorylcholine and o-Phosphorylethanolamine: A Molecular Dynamics Study of Solvation Effects. *Journal of the American Chemical Society* **1994,** *116* (13), 5916-5926.

23. Gaalswyk, K.; Awoonor-Williams, E.; Rowley, C. N., Generalized Langevin Methods for Calculating Transmembrane Diffusivity. *J Chem Theory Comput* **2016,** *12* (11), 5609-5619.

24. Hummer, G., Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations. *New Journal of Physics* **2005,** *7*(1), 34-34.