Surface Assessment Via grid Evaluation

Summary

SuAVE Tutorial: First Steps

If you use SuAVE in scientific publications, please cite Santos et al. SuAVE: A Tool for Analyzing Curvature-Dependent Properties in Chemical Interfaces. 2020. *J. Chem. Inf. Mod*., 60, 2, 473 – 484. DOI: 10.1021/acs.jcim.9b00569

This tutorial covers the main steps to get the most out of the **SuAVE** program. It will guide users through the installation and proper use of the program. **SuAVE** can be compiled in any operational system. The user is advised to refer to the original paper for an in-depth description of the algorithm implemented in **SuAVE**[1]. **SuAVE** is distributed free of charge, and it is maintained by a single developer. If you experience code issues, we kindly request you to contact us at suave.biomat@gmail.com.

1. Installation

- Prerequisites
- cmake installed
- gfortran installed (or any other fortran compiler)
- library libquadmath installed

- Download the code

- From GitHub web Page (https://github.com/SuAVE-Software/source) clone the repository through the following commmand:

  $ git clone https://github.com/SuAVE-Software/source.git

- Dowload the compiled version from SuAVE Web Page (https://www.biomatsite.net/suave-software)

- Compile and install the source code

If you downloaded the compiled version you just need to insert the files on a convenient path in order to be run. If you downloaded the source code, please compile it by the use of any FORTRAN compiler and the MakeFile.

Enter the directory where you have downloaded the code

   $ cd ~/PATH_TO_SRC/

Edit makefile in order to proceed with the installation process. Edit INSTALL_PATH content to update the PATH where you want to place the compiled source, and also the FCFLAGS to adapt it to your needs.

   FCFLAGS = -O2 (DEFAULT)
   INSTALL_PATH = /usr/local/suave (DEFAULT)

(The use of flags -O2 or -O3 is well accepted by this compiler and the code. It will be helpful for extracting the best performance of SuAVE)

Once with the makefile updated, run make!

   $ make
   $ sudo make install

Insert the following directives in the .bash_profile file:

   export SUAVE=/usr/local/suave
   export PATH=$SUAVE:$PATH

Update the bash

   $ source .bash_profile

   or

   $ source .bashrc

2. Brief Introduction to SuAVE

This section describes entry files, routine options and analysis tools for use of **SuAVE**. The first step is the preparation of three entry files required by **SuAVE** to build and adjust the interpolating grid to a given chemical surface. The first file should contain the atomic coordinates of the chemical system in the .**pdb** format. The choice of the **pdb** format allows greater compatibility with different computational simulation programs. This .**pdb** file could contain a single structure or a collection of structures generated via MD or MC simulations, which corresponds to a trajectory file for the system of interest. The trajectory file is not required to contain a temporal evolution of the chemical system, which may be useful to analysis of configurations generated by stochastic algorithms such as MC simulations. An error message will appear if you input the wrong format type.

The second and third .**ndx** files are generated by the *s_index* tool in **SuAVE**. This file should contain the atom number sequence that will be used by the program to generate the interpolation surface and to perform density profile between chemical groups. These steps will be executed consecutively in loop until **SuAVE** reaches the end of the entry file. For each loop, the program reads the atomic coordinates in the .**pdb** file and performs the interpolation of points in the chemical surface to build the surface grid. Thus, each surface fitted grid will represent a given spatial configuration of the system, and will make possible the analyses of geometric properties. The command line options for each **SuAVE** tool can be seen upon typing the tool name in the computer terminal.

After this brief description, special attention should be given to the proper choice of the atoms to be used in the index file. The adequate choice of index atoms is detrimental to the accurate property analysis by **SuAVE**. This is so because the calculated properties depend strongly on the proper fitting of the calculated grid to the chemical surface. The latter, in turn, relies on the correct choice of atoms to be used in the surface point interpolations. The inappropriate choice of index atoms may lead to incorrect representation of the chemical surface configuration, which will yield inaccurate values from the structural analysis.

Some of the desirable characteristics for the proper choice of index atoms are:

1. The chosen atoms must be distributed along the full length of the studied interface;

2. The chosen atoms must be in sufficient numbers to describe well clefts and depressions in the surface that may be relevant for the calculation of the property in consideration;

3. The chosen atoms should ideally be connected to chemical groups with low atomic fluctuation. Highly mobile chemical groups could introduce noise in the analyzed properties.

3. Tutorial

3.1. Construction and Assessment of the Calculated Grid

The *s_index* program builds the index file used in the varied analyses performed by **SuAVE**. For this tutorial, it is provided an input file named *single.pdb* to be used by new users to test different functionalities. The file *single.pdb* contains the atomic coordinates of a lipid membrane. In general terms, the atomic coordinates used to chose the index atoms should represent a continuous structure, whose configuration is approximately flat. This is important to ensure that atoms from different leaflets in the bilayer can be properly distinguished. As a rule of thumb, users are advised to use the initial configuration after geometry optimization as input to select the index atoms. The *s_index* program can be run via the following line command.

s_index -in single.pdb

The command line will list the residues types present in the *single.pdb* file. Select the XYA residue. Sequentially, it will list all atoms present in this residue. Select the phosphorus atom. At the end of the program execution, it will have generated two files *ind1.ndx* and *ind2.ndx* which will contain a numbered list of the 256 phosphorus atoms per monolayer and corresponding to 128 Lipid-A molecules.

It is important to assess whether this atom selection will indeed provide the best grid fitting to the chemical surface representing your system. This can be done via the

program **s_grid**, which not only generates an interpolating grid from the chemical surface using the chosen atoms, but also estimates the root-mean square deviation (RMSD) between each grid point and the index atoms in the chemical surface. The **s_grid** program can be run via the line command

s_grid *-in single.pdb -ind ind1.ndx -bin -rmsd*

The *-bin* flag allows the choice of the number of grid partitions along the x and y-axes. This will define the interpolation mesh resolution constructed on the chemical surface of interest. The *–rmsd* flag specifies the calculation of the distance between the atoms from the chemical surface (used in the interpolation) and the grid surface developed by the program. Using a bin value of 200 (corresponding to 40401 mesh points) the calculated RMSD between the interpolation surface and the phosphorus atoms is approximately 0.1261 nm. This value is smaller than a carbon-carbon bond and indicates that the interpolation procedure developed by **SuAVE** can describe point-to-point of the surface of interest with an average error of up to 0.1261 nm.

The **s_grid** program writes out the *grid.pdb*, *adjust.pdb* and *rmsd.xvg* files, which contain Cartesian coordinates of the interpolating grid, the atomic coordinates of the user-selected index atoms and the RMSD value calculated for each structure in the input file defined by the use of the *–in* flag, respectively. The VMD program (www.ks.uiuc.edu/Research/vmd/) can be used to visualize the atomic coordinates in the *grid.pdb* and *single.pdb* files (Figure 1)[2]. It can be seen that the green interpolating surface encompasses nearly all the phosphorus atoms of the leaflet, and thus accurately delineates the membrane surface shape (Figure 1).
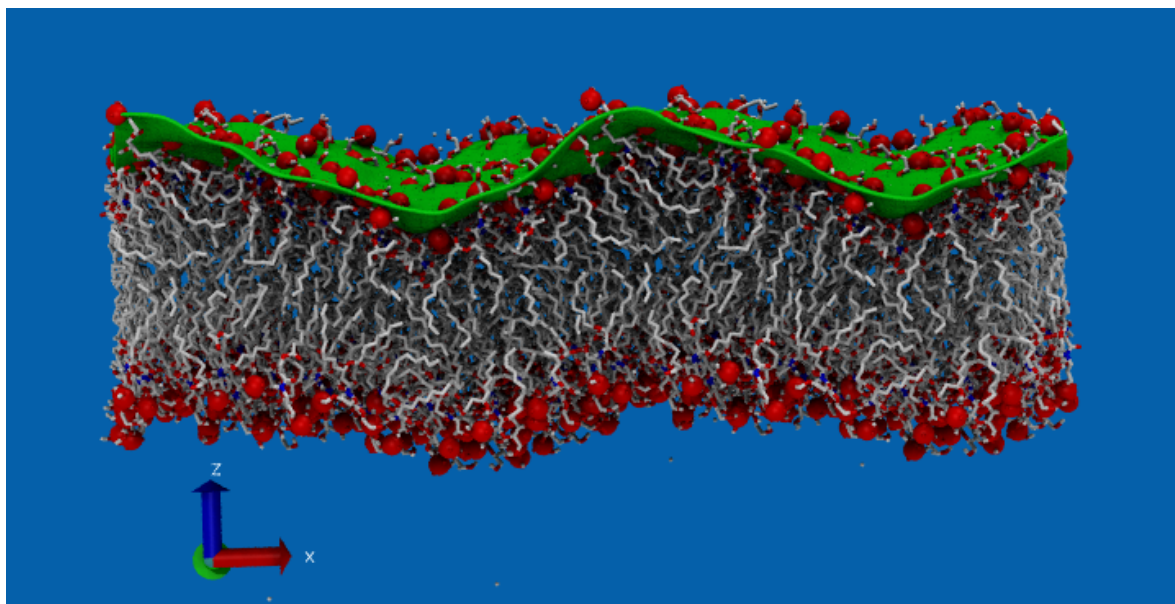
Figure 1. System configuration depicted together with the interpolating surface generated by SuAVE (in green). In red are shown the phosphorus atoms indicated in the index files and in white are the alkyl chains of lipids.

It indicates that the structural analyses can be performed reliably for this system. The visual assessment for the fitting of the calculated grid on the chemical surface and the estimate of the associated RMSD between them should be performed for every system before the calculations of the curvature-dependent structural properties via SuAVE.

3.2. Analyses of Curvature-Dependent Properties
a. Area per lipid

The *s_area* program calculates the lipid interfacial area of a membrane via the command line below

s_area –in traj.pdb –ind1 ind1.ndx –ind2 ind2.ndx –bin –rmsd

When asked for the number of lipids for the Lipid-A bilayer, the user must provide a value of 128, which corresponds to the number of lipids composing each monolayer of the membrane. *s_area* will calculate the surface area and divide it by the number of lipid molecules composing the membrane surface to obtain the interfacial area by lipid or area per lipid. Using half of the mesh resolution described in the index file

check (10201 points, bin = 100), **_s_area_** will generate the _area-med.xvg_ file that can be plotted to yield the data in Figure 2.
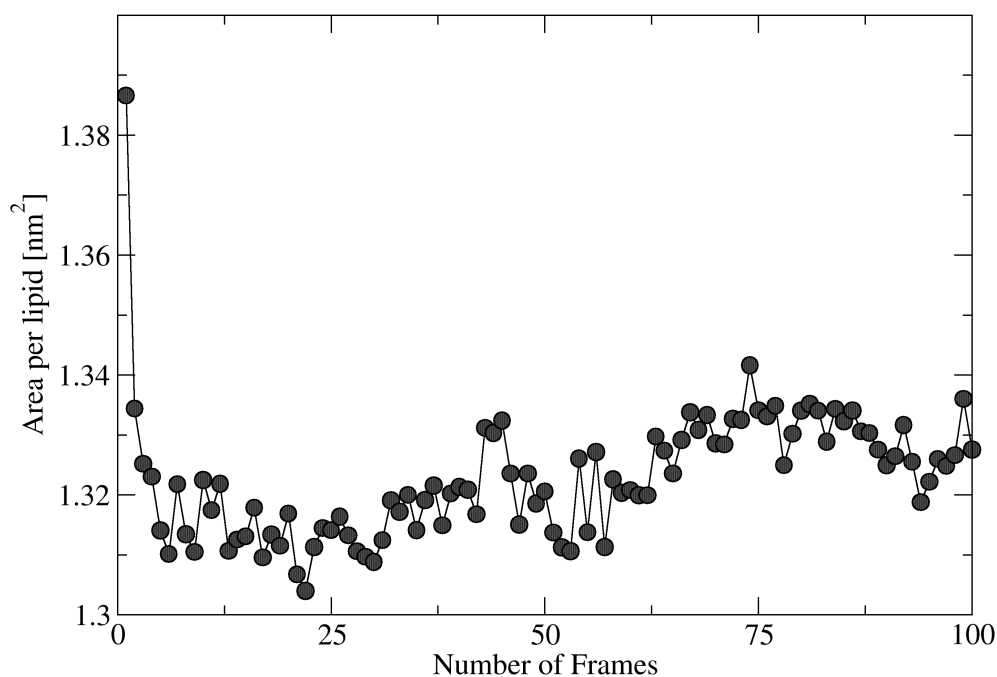

Figure 2. Interfacial area per lipid calculated for 101 trajectory frames.

Experimental estimates of area per lipid for lipopolysaccharide (LPS) membranes in the crystalline liquid phase are of the order of 0.26 nm^2 [3]. This value corresponds to individual alkyl chains in hexa or hepta-acylated LPS molecules. Penta-acylated LPS are estimated to have an area per lipid of ca. 1.30 nm^2, which is fairly consistent with the values estimated by **SuAVE** (Figure 3). In addition to experimental data, computational simulations of LPS with different force fields and from different groups led to similar values of area per lipid [4][5]. It is also important to notice that the trajectory file corresponds to a simulation of 200 ps of the Lipid-A bilayer. This is clearly not long enough for the system to reach equilibration, explaining the significant variation of the area value for the first structure. The calculated RMSD values, written to the _rmsd.xvg_ file, are on average ca. 0.1231 nm. This value combined to that calculated from the index file checking ensures that the calculated grid describes the chemical surface with precision.

b. Bilayer thickness and volume per lipid

The thickness and volume per molecule of a lipid bilayer are calculated using the *s_thick* program using the command line below

s_thick –in traj.pdb –ind1 ind1.ndx –ind2 ind2.ndx –bin –rmsd -range

The number of lipids and the resolution of the interpolating grid should have the same values used in the previous analysis. The *s_thick* program will write the *thick.xvg* and *volume.xvg* files containing the bilayer thickness and the volume per molecule as a function of frame number or simulation time (Figure 3). The flag –*range* defines the minimum and maximum values of the bilayer thickness to be used in the construction of level curves for the graphical representation of this variable (see Figure 4). When asked by *s_thick* the user should enter the values of 2.8 nm and 3.9 nm, respectively.

The calculated thickness values are in agreement with experimental and computational measurements for LPS membranes containing 12 to 16 carbons in their alkyl chains and computational estimates for Lipid-A containing 12 carbons in their longer alkyl chains[3][6]. The calculated RMSD for this analysis is identical to that calculated for lipid interfacial area analysis, i.e. 0.1231 nm. The time-averaged thickness, as well as other properties, can also be presented as topographical maps of the surface with colored level curves representative of variations of this property. The file *thick.xpm* can easily be converted to the eps format using the xpm2ps tool distributed with the GROMACS package [7]. Topographical maps for the Lipid-A bilayer are shown in Figure 4. Topographical maps are useful to illustrate how the thickness varies over the bilayer surface, as for instance, due to the binding of peptides or proteins to the surface. For the Lipid-A bilayer, such changes are minor since the thickness maximum and minimum values are very close, and no specific thickness pattern can be observed on the surface (Figure 4).
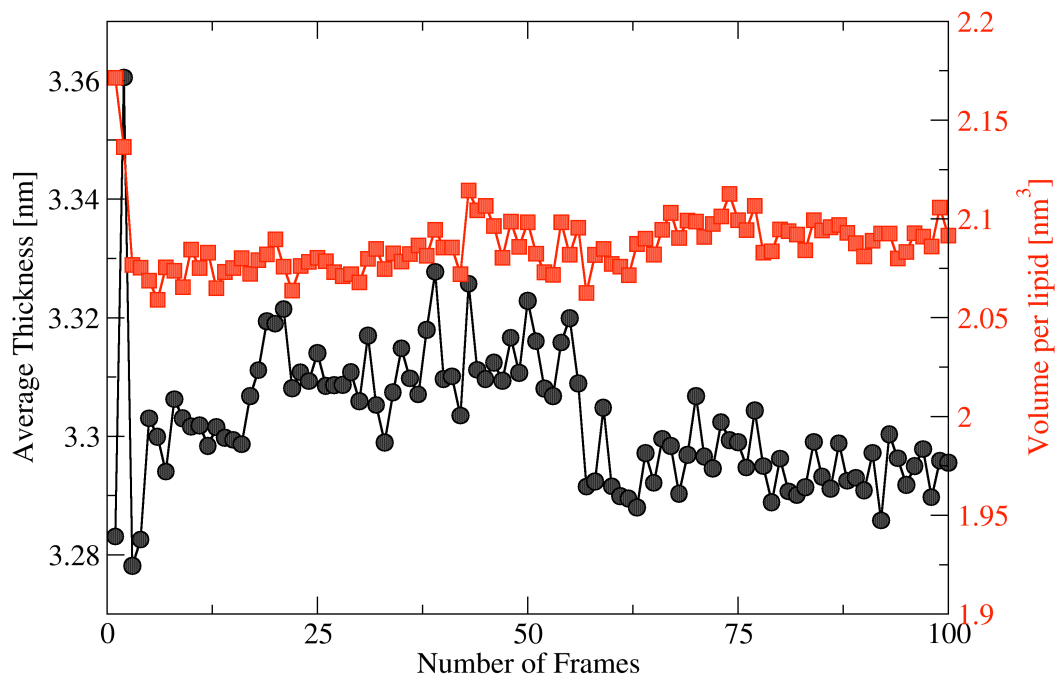
Figure 3. Average lipid bilayer thickness (in black) and volume occupied by each bilayer lipid (in red) calculated for the 101 frames constituting the trajectory.
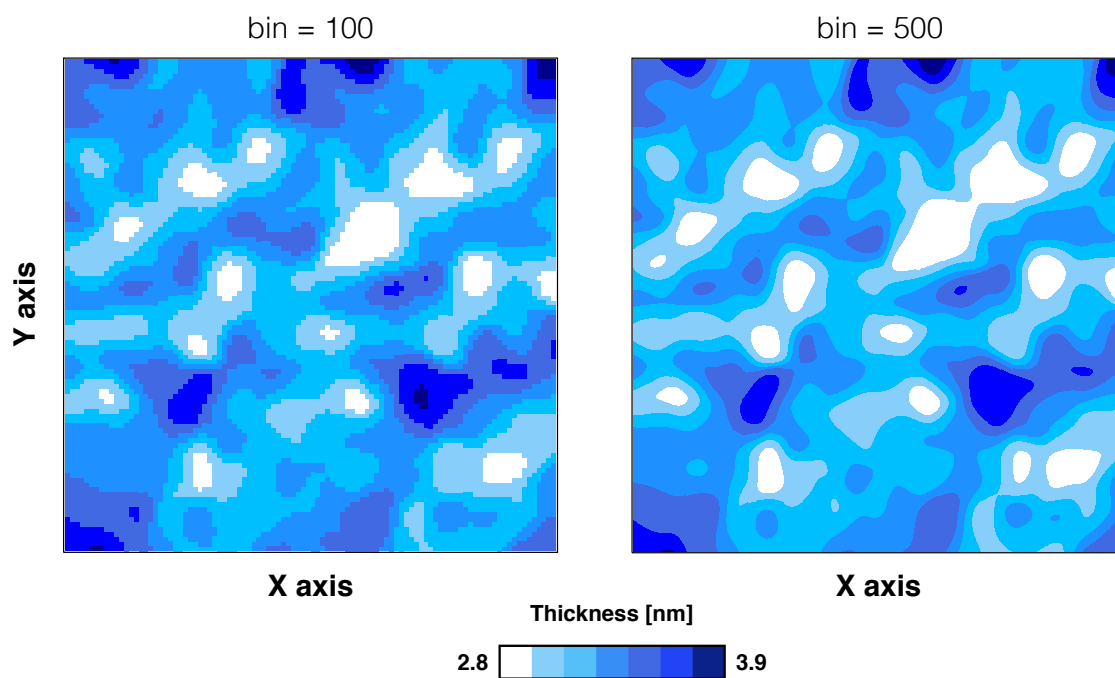

Figure 4. Two-dimensional distribution of lipid bilayer thickness calculated with a mesh resolution of 10201 and 251001 points and corresponding to bin values of 100 and 500, respectively.

c. Surface curvature angle

The *s_order* program calculates the surface curvature angle or the angle between the vector normal to the system surface and each vector normal to the surface of the rectangular partitions [1]. The curvature angle analysis provides a description of the arrangements of lipid aggregates according to their supramolecular morphology and molecular configurations. Overall, planar interfaces are expected to exhibit a very sharp peak with average curvature angles around ca. 15º. On the other hand, surfaces with an expressive angular deflection are expected to present a broad angular distribution and average curvature angles around 30º or higher. These values can vary more or less depending on the chemical nature of the interface. The command line below can be used

s_order –in traj.pdb –ind ind1.ndx –bin –rmsd –range

When asked, the user should enter the values of -0.5 and 1 for the lower and upper boundary assigned to the curvature order parameter limits used to set up the *order.xpm* file [1]. *s_order* will write the file *hist.xvg* containing the distribution of the normal vector angular deviation of the surface in relation to the normal axis of the system (Figure 5). Distribution of the deviation angles of the normal surface vector represents an average over the entire surface. For this reason, surfaces with smooth undulations such as the Lipid-A bilayer used in this tutorial will exhibit an average surface curvature angle around 15 °. This observation is also consistent with the calculated mean value of the curvature order parameter (S_c) written to the file *order_aver.xvg* (Figure 6).
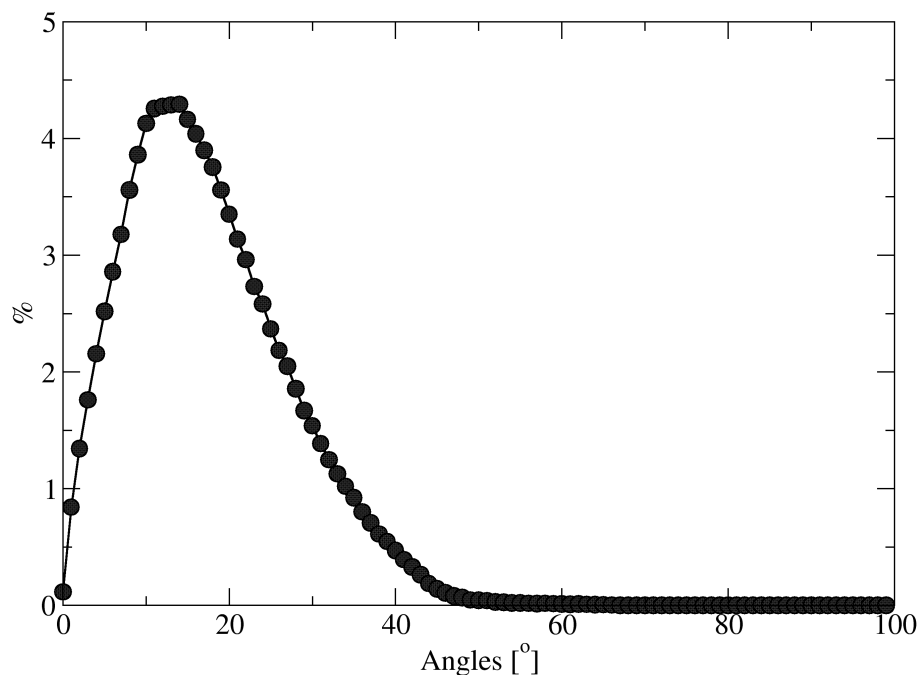
Figure 5. Distribution of the surface angle deviation between the normal vector of the surface rectangular grid partitions with respect to the vector normal to the system for the Lipid-A bilayer.
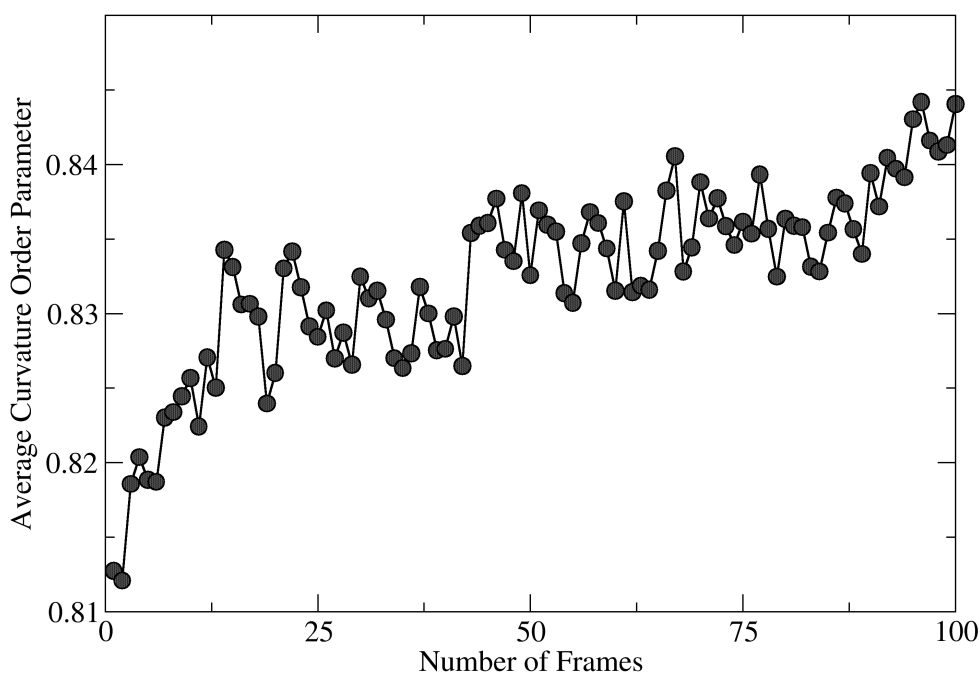

Figure 6. Average values for the curvature order parameter measured for different frames.

The surface curvature order parameter can be represented as two-dimensional frame-averaged distributions (Figure 7). It can help to pinpoint the local where the membrane curvature changes, which can be visually correlated to the presence of peptides, proteins or other molecules on the system surface. (Figure 7). Towards this

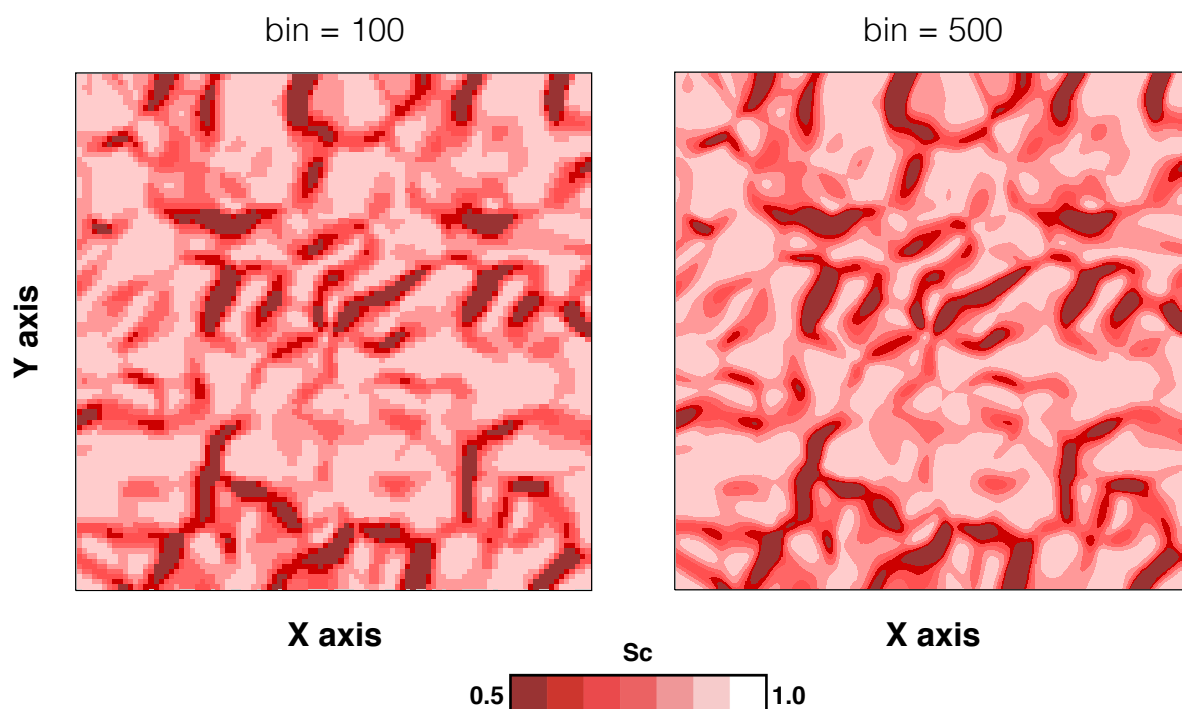end, the file *order.xpm* is written, and can be easily converted to the EPS format as already discussed.

bin = 100   bin = 500

Figure 7.Two-dimensional distribution of the averaged curvature order parameter Sc for the Lipid-A bilayer calculated with a mesh resolution of 10201 and 251001 points, which corresponds to bin values of 100 and 500, respectively.

d. Partial density profile of chemical groups

The calculation of the density profiles by the program **s_index** requires additional index files, *i.e.* index files containing all the atoms for which the density profiles will be calculated. In this tutorial is used the atoms composing the alkyl chains (residue LP1) and the acetyl glucosamines (residue XYA) as well as phosphorus atoms (atom P), counterions (residue CA), and the solvent (residue SOL). The respective index files can be created using the command line below, and selecting the atom groups of interest

*s_index -in single.pdb -residue **LP1** -dens*

The *-residue* flag will list the numbers of every atom in a given group selection, while the *-dens* flag signals to the program that the **.ndx** file will be used for the density profile calculation which, in turn, must be printed in a format suitable for this analysis. The **s_index** program will write an output file, *dens.ndx*, containing 15616 lines with

the numbering of each atom within the selected atom group. This file should be renamed accordingly the atom group it refers to for easier identification. The user is advised to use the lp1.ndx naming for the present tutorial. The same procedure should be repeated for the remaining atom groups by updating the residue or atom names after the *–residue* flag, *i.e.* CA, XYA and SOL and so forth. **s_dens** will calculate the corresponding density profiles using the newly generated atom indexes, via the following command line

s_dens *–in traj.pdb –ind1 ind1.ndx –ind2 ind2.ndx –dens lp1.ndx –bin –rmsd –slices*

The command should be repeated for every atom index selection. The bin value used in the analysis must be the same as in the previous property analyses. It is recommended to use 50 slices. At the end of program execution, a *density.xvg* file will be generated by default containing the density of the selected group in number of atoms per cubic nanometer. To facilitate file recognition, the user is advised to rename it lp1.xvg. The procedure should be repeated for the other groups.

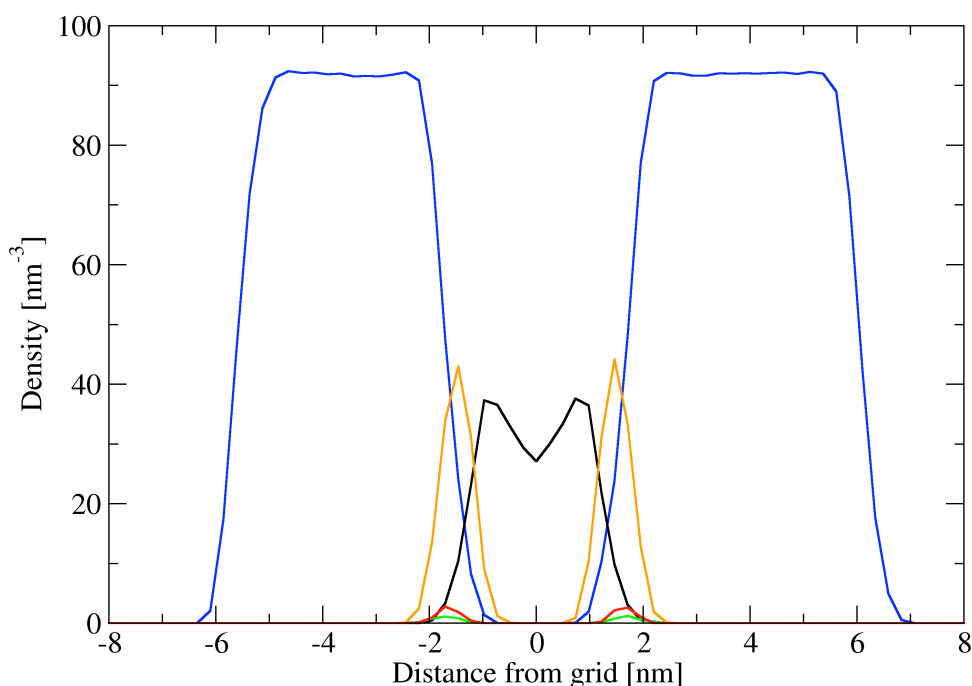The result of joining all density profiles can be seen in Figure 8.

Figure 8. Partial density profile for selected chemical groups composing the Lipid-A system: alkyl chains (black), solvent (blue), counter ions (green), acetyl glucosamines (orange) and phosphorus atoms (red).

e. Other tools

SuAVE is distributed with additional analysis tools to assist in the observation of structural modifications of a chemical surface. *s_grid* is a program that writes the interpolation grid calculated by SuAVE for each frame of the given trajectory. Thus, *s_grid* generates a "trajectory of grids" corresponding to each frame in a given trajectory file useful to verify how the grid interpolation changes throughout the simulation. The command to run this tool is

s_grid –in traj.pdb –ind ind1.ndx –bin

It is also possible to analyze the topography of the chemical surface by using the *s_topog* program (Figure 9). It builds two-dimensional maps describing the level surfaces from the interpolation grid points. The command to run this tool is

s_topog –in traj.pdb –ind1 ind1.ndx –ind2 ind2.ndx –bin
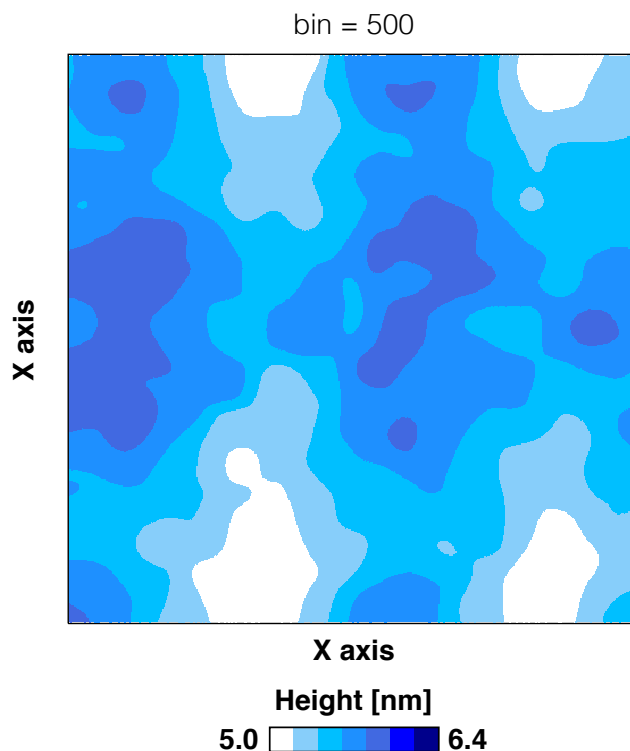
bin = 500

Figure 9. Level curves built from averaging the interpolation grid points corresponding to the frames from the MD simulation of the Lipid-A bilayer used in this tutorial.

4. Cited Literature

[1] Santos, D. E. S., Pontes, J. F. S., Lins, R. D., Coutinho, K., Soares, T. A. SuAVE: A Tool for Analyzing Curvature-Dependent Properties in Chemical Interfaces. 2020. J. Chem. Inf. Mod., 60, 2, 473–484.

[2] Humphrey, W., Dalke, A. and Schulten, K., `VMD – Visual Molecular Dynamics', 1996. J. Molec. Graphics, 14.1: 33-38.

[3] Snyder, S., Kim, D.; McIntosh, T.J. Lipopolysaccharide Bilayer Structure: Effect of Chemotype, Core Mutations, Divalent Cations, and Temperature. 1999. *Biochemistry,* **38**: 10758-10767.

[4] Kim, S.; Patel, D.S.; Park, S., Slusky, J., Klauda, J.B.; Widmalm, G., Im, W. Bilayer Properties of Lipid A from various Gram-Negative Bacteria. 2016. *Biophys. Journal*, **111**: 1750-1760,2016.

[5] Nascimento, A.J., Pontes, F.J.S., Lins, R.D., Soares, T.A. Hydration, ionic valence and cross-linking propensities of cations determine the stability of lipopolysaccharide (LPS) membranes. 2014. *Chem. Comm.*, **50**: 231-233.

[6] Dias, R.P., da Hora, G.C.A., Ramstedt, M.; Soares, T.A. Outer Membrane Remodeling: The Structural Dynamics and Electrostatics of Rough Lipopolysaccharide Chemotypes. 2014. *J. Chem. Theory Comp*, **10**: 2488−2497.

[7] Van Der Spoel, D., Lindahl, E., Hess, B., Van Buuren, A. R., Apol, E., Meulenhoff, P. J., Tieleman, D. P., Sijbers, A. L. T. M., Feenstra, K. A., Van Drunen, R., Berendsen, H. J. C. Gromacs User Manual version 4.5.6, 2010.