Vesicle Viewer: Online analysis of small angle scattering from lipid vesicles

Aislyn Lewis-Laurent,[†] Milka Doktorova,[‡] Frederick A. Heberle,[*,¶] and Drew Marquardt[*,†,§]

[†]*Department of Chemistry & Biochemistry, University of Windsor, Windsor, ON, Canada*

[‡]*Department of Molecular Physiology and Biological Physics, University of Virginia School of Medicine, Charlottesville, Virginia*

[¶]*Department of Chemistry, University of Tennessee, Knoxville, Tennessee*

[§]*Department of Physics, University of Windsor, Windsor, ON, Canada*

E-mail: fheberle@utk.edu; drew.marquardt@uwindsor.ca

Abstract

As progress is made toward suitable experimental setups to investigate biological membranes via small angle scattering (SAS) techniques, SAS users will generate larger volumes of data. Much of the critical information inside these data-sets is not easily accessible to researchers, who have limited time to analyse results by hand or may lack the necessary scientific background to process such data. Easy to use data visualization software can allow them to take full advantage of their SAS data and maximize the use of limited resources. In this project, we developed an internet-based application, called Vesicle Viewer, to visualize and analyze small angle scattering data generated in the study of lipid bilayers. Vesicle Viewer models SAS data using the EZ-SDP model. In this way, key bilayer structural parameters, such as area per lipid and bilayer thickness, can be easily determined. This application primarily uses Django, a python package specialized for the development of robust web applications. In addition, several other libraries are used to support the more technical aspects of the project – notable examples are MatPlotLib (for graphs) and NumPy (for calculations). Without the barrier of downloading and installing software, the development of this web-based application will allow scientists all over the world to take advantage of this solution, regardless of their preferred operating system.

Introduction

X-ray and neutron scattering data have a wide range of applications in the work of biochemists and biophysicists, but remain complex to process and interpret. While models are available (including the simplified scattering density profile model), applying these models is impossible to do by hand. In the absence of a universal, standardized tool, researchers are forced to build their own solutions. There are several reasons why this is a less than optimal approach.

Researchers are extremely busy, and often do not have time to build robust solutions. This means the computational tools that they create are optimized for speed of development rather than ease of use or consistency. This can often lead to solutions that are only viable for the person who programmed them. New team members, students (grad or otherwise), or other research groups cannot take advantage of this hard work. Data is often left to sit until the developer has time to handle it, which could be an extended period of time.

Additionally, these solutions may not be prepared with future projects in mind. This means even if more advanced features, additional calculated parameters or new models would be of use, they may not be within reach of the team. This again leads to data not being tapped for its full potential with new discoveries and advancements being left on the table.

With the advent of the scattering density profile (SDP) model for the analysis of lipid bilayer structures, the demand for accessible small angle X-ray scattering (SAXS) and small angle neutron scattering (SANS) has increased. The SDP model approach was crafted by

Wiener and White[1,2] and perfected by the efforts of Kučerka and co-workers[3–11] and addresses the complementary aspects between SAXS and SANS by jointly refining the data sets from both techniques to produce scattering density profiles.

Despite the utility of the SDP approach, it is technically difficult to implement (as it requires complex computer coding) and is limited in scope that it can be used due to over parameterization. For this reason we have developed and implemented a smeared 6-slab model we refer to as EZ-SDP, which has been utilized previously.[12,13] The EZ-SDP approach requires only two varying structural parameters, namely the area per lipid and the headgroup thickness (discussed further). The strategy of modeling a lipid bilayer as a series of slabs or strips is not a novel approach;[14] however, our EZ-SDP approach does facilitate the determination of volume probabilities for key structural groups using experimentally determined volumes (Equations 1-5 Supplementary Information).

In a bid to make small angle scattering (SAS) data analysis easier for users and reduce group-to-group variation, we have developed the Vesicle Viewer online platform which implements the aforesaid EZ-SDP model. We have tested this platform with SAS data generated from molecular dynamics simulation for a single lipid bilayer, a multi-component symmetric and asymmetric lipid systems. We further evaluate experimental data of 1-stearoyl-2-docosahexaenoyl-sn-glyerco-3-phosphocholine (SDPC) with Vesicle Viewer and directly compare to previously reported results derived from SDP analysis.

Methods and Materials

Vesicle Viewer Models

The EZ-SDP approach determines the key structural features most desired in membrane studies from a set of known parameters and two varying parameters for the symmetric model (Eqn 2) and four varying parameters for the asymmetric bilayer model (Eqn. 6 Supplementary Material). Table 1 summarizes the experimentally fixed and the varying parameters for the two models, which include: area per lipid (A_L), the overall bilayer thickness (D_B), the bilayer hydrocarbon chain thickness ($2D_C$), and the distance between the maxima of the electron density profile (D_{HH}).

$$\begin{aligned} F(q) \quad = \quad & \frac{2e^{-(q\sigma_S)^2/2}}{(qD_H A_L V_T V_W (V_C - 2V_T))} |V_T[b_W(A_L D_H - V_H)(V_C - 2V_T) \\ & + V_W b_H (V_C - 2V_T) - V_W A_L D_H (b_C - 2b_T)] \sin\left(\frac{(qV_C)}{A_L}\right) \\ & + V_T (V_C - 2V_T)(b_W V_H - b_H V_W) \sin\left(\frac{qD_H + (qV_C)}{A_L}\right) \\ & + V_W A_L D_H (b_C V_T - b_T V_C) \ \sin\left(\frac{2qV_T}{A_L}\right) | \end{aligned} \tag{1}$$

Table 1: Relevant Paratemers in the EZ-SDP Model

Parameter	**Descriptions**
	Experimentally Derived
V_C	Lipid hydrocarbon chain volume [Å^3][1]
V_H	Lipid headgroup volume [Å^3][1]
V_T	Lipid terminal methyl volume [Å^3][1]
V_W	Water volume [Å^3]
	Experimentally Derived
b_C	Lipid hydrocarbon chain scattering length[2]
b_H	Lipid headgroup scattering length[2]
b_W	Water scattering length[2]
	Varying Parameters
A_L	Area per lipid [Å^2]
D_H	Lipid headgroup thickness [Å]
σ_S	Smoothing parameter for slab model

[1] Can be varied in Advanced Options

[2] [fm Å^{-3}] for neutrons and [e^- Å^{-3}] for X-rays

*Bilayer parameters double for the asymmetric model. The inner and outer leaflet parameters are signified by '^{in}' and '^{out}' respectively.

The models allow for the A_L and D_H to be refined; the lipid volumes can be optimized in the *Advanced Options* of the program. Other key structural parameters that are desired by researchers are subsequently calculated from the optimized A_L and D_H; most notably, the bilayer thickness (D_B, Eqn. 2) and the hydrocarbon thickness ($2D_C$, Eqn. 3). Further, the models yield a commonly used headgroud-headgroup distance (D_{HH}) which is calculated from the headgroup probability distribution peak-to-peak distance determined from the optimized fit parameters. Upon testing, no measurable discrepancy appeared between the D_{HH} values calculated by Vesicle Viewer and those measured in the traditional method of using the location of the electron rich phosphate groups from an electron density profile.

A set of volumes are required for model fitting; these are experimentally derived. There are a number of strategies that have been used to measure the lipid volumes including density measurements and neutral flotation method.[15] The total lipid volumes (V_L), headgroup volumes (V_H) and hydrocarbon volumes (V_C) are summarized in the work of DiPasquale *et al.*.[16]

$$D_B = \frac{2V_L}{A_L} \tag{2}$$

$$2D_C = \frac{2V_C}{A_L} \tag{3}$$

Vesicle Viewer Description

The purpose of the Vesicle Viewer project is to make established mathematical models easily accessible to researchers. This may increase the efficiency of their workflow, and ultimately increase the utility of newly collected and existing SAXS and SANS data. The main focus of this project is to build both a well optimized algorithm for producing results and a carefully constructed and easy to understand user interface. Figure 1 showcases this - important parts of the interface are clearly labelled and well organized, ensuring that using the tool is intuitive. Source code can be found at GitHub (https://github.com/AislynLaurent/vesicle-viewer) along with a listing on Zenodo (10.5281/zenodo.4653420).

Less focus was placed on making the system highly flexible, though several features were put in place specifically to cater to more advanced users. This necessitated some restriction on the user's ability to fine tune the mechanisms used in fitting, as well as the ability to fit for highly specialized samples. These compromises were minimized, however, to ensure that the widest possible audience is able to take advantage of the tool.

Users will input a currently unlimited number of data-sets to be jointly optimized using the EZ-SDP model. A least squares approach is used to optimize a line of best fit to each data-set, providing the user with a visual representation of the fit, fit statistics, and a final set of fit parameters. Fitted values are automatically stored, and the user is free to continue to process the results or to export them for further work using some other software. Specific inputs and outputs are outlined below.

User Input - Overview

Users will have to input various information during different stages of data processing. Initially they are required to provide identifying authentication information, used on the back end for both security and organization. This includes a username, email address, and institution name. Additionally, each user must agree to basic terms and conditions. This provides both the user and the developer with clear roles and responsibilities, for example maintaining user privacy and citing and sourcing results.

Afterward several layers of organization allow users to separate results and prevent repeated processing of the same data. Projects contain high level information like the model used (symmetrical or asymmetrical) and the temperature. Samples contain more detailed information, like variations in lipids, data-sets and fit parameters. Each of these data elements can be edited at any time. These changes are not retroactively enforced, but are reflected in any new calculations performed.

The fundamental data structure is the fit parameters. These are initially populated with suggested values. Some parameters must be calculated, and therefore are only presented to the user after the initial generation of a parameter set. This can also be tweaked allowing advanced users fine control over calculations.

Each of these data elements is stored in the database until the user makes the decision to edit or delete it. This allows for quick retrieval and increased efficiency. Vesicle Viewer uses the PostgreSQL database architecture because it is robust, flexible and efficient.

Calculated Output

Volumes are interpolated from temperature-dependant experimental data[4–8,10,11,16] and are only displayed to the user after a parameter set is generated. Though it is possible for the user to input all the fit parameters by hand, this would require significant prepossessing and potentially prevent new or inexperienced users from taking advantage of the tool.

Once a parameter set has been generated, it is fit simultaneously to all provided data-sets. A graphical representation of the fit is displayed for visual analysis. This allows the user to quickly assess the quality of fit. More detailed statistics are also provided, including the initial and final values for each parameter, the reduced chi-squared, and the number of iterations required to fit.

The main output of interest are the final values for the lipid area (A_L), terminal methyl volume (V_{TM}) and the head-group thickness (D_H). These are optimized during fit and can be used in future work. All of this information, in addition to the calculated values for intensity of each data-set can be exported as a CSV file which can be further processed using programs like Microsoft Excel.

Calculated output values are also stored in the database so that they can be referenced later on, or tweaked and re-fit when required. It is important to note the Vesicle Viewer is not intended to permanently store any information, and users are encouraged to export any results they intend to keep.

Python Libraries

Vesicle Viewer employ several Python libraries to perform the data processing and analysis mentioned. Many of these are standard Python libraries, like CSV, and do not provide noteworthy functionality (a full list is available as part of the code posted on GitHub.com). Below is a list of the libraries used, the functionality they provide and the reason(s) for selecting them.

- Django
  - A web development platform that allows python code to be run alongside HTML, CSS and JavaScript. Provides necessary core functionality to run calculations online.
  - While other, more minimal platforms like Flask may have been workable, Django is a more robust and well supported option.
- matplotlib [17]
  - A graphing library with support for a large variety of plotting and visualization options. Allows for adjustment of labeling, line style, scale and more.
  - Chosen because it is a well supported and robust library, and because it connects seamlessly with mpld3.
- mpld3
  - Converts graphs generated by matplotlib into HTML, CSS and JavaScript so that they can be displayed on a web interface.
  - Allows for fully featured graphs (including zoom functions) to be displayed in the web-app simply and easily.
  - Other options with similar functionality required the use of 3rd party tools.

- numpy [18]
    - Mathematical library containing advanced operators and array manipulation modules. Seamlessly integrates with other libraries used in the project like matplotlib and LMFit.
    - Though Python has a good variety of mathematical functions built in, numpy improves the work-ability of arrays significantly. Additionally, several required functions are not part of the standard library, notably trigonometric functions.
- LMFit [19]
    - A wrapper around the SciPy library module ”minimizer” which performs a least squares minimization using the Levenburg-Marquardt algorithm by default (other minimization algorithms are available).
    - While all the necessary functions are present in SciPy, their use is much more streamlined in LMFit. Specifically, the built-in functionality allows the user to specify parameters, their upper and lower bounds, lock or allow them to vary, and much more.

Workflow

Figure 2 shows a high level description of the workflow in Vesicle Viewer. Several layers of processing occur. These stages include user authentication, user input, data pre-processing, fitting, visualization and output.

User Authentication

Data is organized at the highest level by ”owner”. This makes data private to each individual user. As such, each page must authenticate the user before rendering it’s contents. This is done by collecting the user identity at each page - if the current user is known by the system, their data is displayed.

Users must sign up for an account through a standardized system, made available by Django, which takes their name, email address, institution and password. Django's built-in systems automatically encrypt their password so that the information is not directly stored in the database. Once users are ready to begin work, they login using this same information, allowing each page to check their identity.

User Input - Detail

Once authenticated, users are provided a list of their current projects and custom lipids. Projects are the second highest level of organization. To create a project, users select the model they would like to use (symmetric or asymmetric), the temperature of their system, and the lipids present in their system. They can also choose to enable advanced options (e.g. varying normally fixed parameters).

After setting up their project, users can create samples. This additional layer is entirely organizational, and is designed to prevent the repeated entering of duplicate information. At the sample level, more detailed information is collected about the sample composition, including the mol fraction of each lipid and, in the case of asymmetric systems, their leaflet location (i.e. inner or outer).

It is at this point that users can select from a number of augmentations to the nominal forms of the lipids they have added. A database of common options will be available, preventing new users from having to manually search for values. Users can specify anything from added deuterium to heavy metals without sifting through long database tables. More advanced users can enter their own custom augmentations by providing the net scattering length change for each section of the lipid. While composition information may be more intuitive for chemists to work with, this approach allows for maximum flexibility.

An unlimited number of data-sets can be added. More details about how data is stored and used will be discussed in the following sections. Users can specify the experimental data type (neutron or X-ray), the proportion of D_2O and data-set name.

Lastly, users generate a parameter set. As with data uploads, some calculations are required at this step. The user is supplied only with parameters which they may wish to specify, like the area per lipid. Suggested values are supplied so that inexperienced users can still perform analysis without a detailed understanding for what reasonable starting values should look like.

Data Pre-Processing

In order to maintain a clean and tidy user interface, several operations are hidden in the back-end of the application. These range from simple calculations to complex data processing. The first of these is hidden behind the data upload step involved in setting up samples.

When the user chooses to upload a data file, the file is first checked for type and size. Afterward, it is read line-by-line, and three arrays are created based on the extracted values. The header and footer are both discarded, as they do not carry usable information to the program. The first column of data is stored as the q value array, the second as experimental intensity values and the third column is assumed to be error values (if no error column is present, this array is filled with 1 values). Any additional columns are discarded. These arrays are the only part of the data file kept in the database, as they are the only information required later in the process.

Once the lipid composition has been set, users can generate initial parameters for their fit. For unknown values, an estimate is populated as described above. Head- and chain-group volumes are calculated based on a volume equation stored in the database for each lipid. The project temperature and the mol fraction of each lipid is inputted, and a combined volume value for the parameter set is calculated. For custom lipids, the user is not able to enter a volume equation for security reasons (some volume equations will require access to python functions). As such, they can add the correct lipid volume for the appropriate temperature at the project level.

Fitting

LMFit is used to perform the fit. LMFit relies on a least-squares minimization approach, and offers a number of convenient features. The steps taken to produce a line of best fit for all models will be described here. More detailed information on the least-squares method used by LMFit can be found in the documentation on their website (https://lmfit.github.io/lmfit-py/).[19]

1. The fit function is called by the user by hitting the "fit" button. This sends the current sample and parameter instances to a separate module for processing. Existing parameters are saved in the database at this stage to allow the user the option to go back to older fits.

2. Parameters stored in the database are converted to LMFit parameters. For each parameter value, the database also automatically stores the upper- and lower-bounds, as well as a boolean value indicating whether the user will allow that parameter to vary or not. These are converted also.

3. "b values" are calculated by checking for lipid augmentations and the mol fraction of D_2O for the data-set. These are added to the overall parameter set.

4. The minimization function is called and transfers the converted parameters along with an objective function which is used to calculate the residuals for each data-set at each iteration.

5. The objective function calls a calculation function which evaluates the appropriate model for each data-set with the current parameters (i.e. the current iteration of the minimization). Each residual is scaled by the error for that data-set and concatenated to a larger array which will eventually be checked for the quality of fit. The water

volume probability (SI: Eqn 5) is also calculated by the objective function and a penalty is added for results where the value dips below zero.

6. Residuals for the current iteration are evaluated for goodness-of-fit. The minimization function decides whether or not to continue or to stop, based on its own criteria.
7. The final iteration is passed back to the "fit" module, where the newly fitted parameters are stored in the database and displayed for the user. The new parameter set is automatically named for the current time.

Data Visualization

Scatter plots are generated for each data-set and displayed for the user before fitting takes place. At this stage their main utility is for the user to optimize the fitting parameters prior to fitting.. Each graph is generated iteratively by matplotlib and rendered into HTML and CSS by mpld3.

After fitting takes place, new parameter values are displayed. The line of best fit is generated by piping these parameter values into the mathematical model used in the fit. The program decides whether or not to display a line of best fit by checking if a set of fit statistics exists for a particular parameter set. This allows the line of best fit to automatically display any changes users make to parameters while also preventing duplicate database entries.

In addition to the data scatter plots and lines of best fit, the program displays fit statistics, volume probabilities and density profiles. Fit statistics are stored in the database so that they can be displayed and downloaded during the output stage. Like the line of best fit, probability and density profile graphs are generated based on the current set of parameters.

Output

Users are able to download a variety of data for further processing. CSV files are used as they are easily processed by multiple different software applications - most notably Microsoft

Excel. Additionally plain text files are compact and require relatively little back-end processing to produce. Vesicle Viewer has the ability to automatically generate two different output files:

1. A "Fit" file which contains the current parameter set, fit statistics like the chi squared value and parameter variance, as well as calculated and experimental I(q) and error values for each data-set.

2. An "SDP" file containing the values for volume probabilities and separate and combined scattering density profile values.

Between these two files, all useful information displayed to the user as part of the interface is made available for them to download and work with as they please.

Molecular Dynamics (MD) simulations

We constructed and simulated two symmetric and one asymmetric bilayers, as summarized in Table 2. The simulation trajectories for the two symmetric bilayers were taken from.[20] The asymmetric bilayer was built and simulated in a way similar to the symmetric bilayers as described in.[20] Briefly, the bilayer was constructed with the CHARMM-GUI web server in 2015.[21–25] The system was simulated with NAMD version 2.7[26,27] and the CHARMM36 force field for lipids and ions.[28,29] Since at the time of construction CHARMM-GUI did not provide an established minimization protocol, the bilayer was first energy minimized for 10,000 steps, then simulated for 500 ps with a 1 fs time-step. Since no constraints were imposed during these initial equilibration steps, in 2% of all lipids in the bilayer (2 POPC, 4 POPE and 2 POPS lipids) the cis double bond between carbons 9 and 10 on the sn-2 chain changed its isomerization to trans and remained like that for the entire duration of the simulation. The production run was performed at a constant temperature of 25 C (298 K) and constant pressure of 1 atm under semi-isotropic pressure coupling conditions. The simulation parameters were the same as the ones used for the production runs of the two symmetric bilayers as described in.[20]

The number density profile for each bilayer was calculated for every atom in the system with the Density Profile tool in VMD[30] at a resolution of 0.2 A. In the calculation of the scattering form factors from the simulation, a hydrogen was changed to deuterium by a matter of changing its label and using the appropriate scattering length.

Table 2: Summary of MD bilayer composition

Bilayer	**Lipid composition**		**Number of lipids**		**Number of**	**Simulation**
	top system	**bottom**	**top**	**bottom**	**waters per lipid**	**time**
1*	POPC	POPC	208	208	45	520 ns
2*	POPE/POPS 70/30	POPE/POPS 70/30	210	210	82	690 ns
3*	POPC	POPE/POPS 70/30	191	220	60	620 ns

* Simulation trajectories were taken from ref [20]

Neutron and X-ray scattering intensity from Molecular Dynamics Simulations

First, an atomic number density profile from the simulation trajectories by bining the z-coordinates of the different atoms to create a histogram. Is is repeated for all frams in the simulation. The histogram is normalized to the number of MD frames processed. Neutron scattering length density (NSLD) profiles and electron density (ED) profiles are generated from the atomic number density by assigning the appropriate scattering length or number of electrons to the specific atoms and divide by the bins volume. For example, the hydrogen atoms associated with the palmitoyl chain of POPC would be assigned the scattering length of deuterium for the chain deuterated POPC (dC-POPC). Finally, SANS and SAXS intensity curves are generated from the NSLD for SANS, and ED for SAXS via the transform:

$$I(q) \propto \int (\rho(z) - \rho_s) e^{iqz} dz \qquad (4)$$

where q is the scattering vector, and $\rho(z)$ and ρ_s are the scattering length or ED along the z-direction of the bilayer and the solvent respectively.

Experimental Data

The EZ-SDP model has been applied to fairly ubiquitous phospholipid systems in past works;[12,31] therefore, we utilize the polyunsaturated fatty acid containing SDPC that has recently undergone the rigors of the full SDP analysis. The SANS and SAXS data were provided from the work of Marquardt et al..[10] The X-ray data were taken at the Cornell High Energy Synchrotron Source (CHESS) G-1 station and the neutron scattering experiments were performed at the High Flux Isotope Reactor (HFIR) CG-3 Bio-SANS instrument located at Oak Ridge National Laboratory (ORNL). Detailed sample preparation and instrument condition are summarized in ref.[10]

Results

Simulated Data

The robust way to assess a new analysis technique or platform is to evaluate data that has an absolute "correct" answer. Scattering data calculated from MD simulations provide "experimental" data where the structural parameters of the bilayer are strictly known. In this way we test whether Vesicle Viewer and our EZ-SDP model yield structural parameters of high fidelity.

Firstly, we examine the simple single lipid case of POPC. Figure 3a illustrates POPC bilayer broken down into its respective slabs and how these slabs correlate to the functional group volume probabilities and ultimately the VV fits of the MD generated scattering data. Table 3 highlights key structural parameters determined by VV and those extracted from the MD simulations. It is particularly note worthy that all the values are within 1 Å (or Å^2) of each other and their variance is less than 5%.

Secondly, we test VV with a symmetric lipid mixture of POPC:POPE:POPS. It is clear that VV captures the structural parameters with an accuracy of <1.3% (Table 3. Notably, the optimized A_L is <0.2% accurate.

Finally, we tested VV on an asymmetric system having dC-POPC on the outer leaflet and a mixture of POPE:POPS on the inner leaflet. Having one of the leaflets deuterated assists with contrast and is most representative of experimental structural studies of asymmetric membranes.[31–34] Despite the complexity of this lipid system all the optimized values are in good agreement with the actual values extracted from MD, with the most notable deviation occurring in the A_L parameter. We discuss the origin of minor deviations below.

Table 3: Comparison of EZ-SDP executed through Vesicle Viewer with those directly from MD simulations

	POPC		POPE:POPS		dC-POPC^{out}/(POPE:POPS)^{in}	
Parameter	**VV**[1]	**MD**	**VV**[1]	**MD**	**VV**[1]	**MD**
A_L (Å^2)	63.9	64.3	55.6	55.5	66.9/54.1	64.2/55.7
D_H (Å)	8.3	8.8	8.1	8.0	9.5/7.8	8.9/8.0
D_B (Å)	39.1	38.4	41.88	42.2	18.7/21.7	19.3/20.9
D_{HH} (Å)	37.2	38.2	41.2	42.0	18.6/20.8	19.0/20.8
$2D_C$ (Å)	28.79	28.8	32.8	32.9	13.8/17.0	14.5/16.4

[1] Implemented using the Vesicle Viewer online platform

Experimental Data

We then apply experimental scattering data from SDPC bilayers to Vesicle Viewer. These have previously been characterized using home-source code for the SDP model and the previous SDP analysis will serve as our benchmark (Table 4).[10] Figure 4 depicts the experimental SAS data for the SDPC as well as the optimized scattering density profiles. Due to the incoherent background associated with such a high hydrogen content in the solvent, the 50% D_2O data appear cut-off at q=0.15 Å^{-1} due to the increased incoherent background associated with increased quantities of H_2O and the fact that values on a logarithmic scale must be $\geq$0.

Table 4: Comparison of EZ-SDP executed through Vesicle Viewer with the full SDP model of SDPC at 30 °C[10]

Parameter	**EZ-SDP**[1]	**Full-SDP**
A_L (Å^2)	69.6 (1.3)	70.4 (1.4)
D_B (Å)	39.3 (0.7)	38.8 (0.8)
D_{HH} (Å)	34.5 (0.6)	35.2 (0.7)
$2D_C$ (Å)	30.06 (6)	29.7 (0.6)

[1]Implemented using the Vesicle Viewer online platform

Discussion

Our Vesicle Viewer web platform utilizing an EZ-SDP model demonstrates the ability to accurately recover bilayer structure from both simulated and experimental data. Although the full SDP model has proven useful in single lipid systems, it becomes an "ill posed problem" for mixtures and asymmetric bilayers. For example, the asymmetric SDP model employed in Eicher et al. has 12 paratemeters that require optimization.[35] This necessitates the establishment of an equally robust, yet simple enough model as to not over-parameterize. Further, the work demonstrates that a joint refinement of a slab model yields the same results (with in error) as the SDP.[35]

Although VV sufficiently captures the bilayer structures from the MD simulations, the very minor deviations from our modeled determined values and those explicilty from the simulation can be explained. The lipid volumes used in the EZ-SDP model were experimentally determined and are stored within the VV database, while the lipid volumes from MD differ from these experimental. For example, our experimentally determined database value for POPC was 1251.5 Å^3 whereas the MD yield a volume of 1234.1 Å^3. Although this is a small difference, it manifests itself largely in the value of A_L as notably mentioned for the asymmetric bilayer. Such discrepancies between In practice, experimentally determined volumes are necessary for the EZ-SDP model; thus, we do not manipulate our volumes to improve the results (although the program is capable) in order to highlight the robustness of our approach.

Developing tools designed with the user experience in mind can therefore be very beneficial for the field as a whole. Having a dedicated development team who can focus on writing clean, modular and optimized code means that tools are easy to use, often updated and can thus produce consistent results. Errors can be addressed quickly and new functionality can continuously be developed. Teams worldwide can also contribute to the project, providing their own feedback on the tool and offering suggestions on how it can be improved.

New users can easily pick up on the workflow of using the tool and process data with little oversight, meaning that data no longer need to wait for a qualified researcher to have time to process them. If a tool is sufficiently easy to use, data can be handed to students with little to no experience and be processed quickly and easily. This type of task delegation could increase the efficiency of teams, leaving tedious work off the desks of busy, highly skills individuals, meaning they can focus on other more high priority tasks.

Vesicle Viewer achieves exactly this solution. Because the tool is focused on providing an easy to understand interface, it can be used as everything from a teaching aid to a computational tool. This, along with step by step tutorials and guides, means any individual with scattering data to process can pick it up in just a few minutes. Researchers can focus their time on their projects and need not be concerned with the nuances of computer programming, data structures or even the mathematical model itself.

Acknowledgement

The authors thank Michael H.L. Nguyen for his critical review and valuable feedback. This work acknowledges support from the Natural Sciences and Engineering Research Council (NSERC) of Canada (funding reference number RGPIN-2018-04841) (D.M.). F.A.H. acknowledges support from National Science Foundation (NSF) grant No. MCB-1817929.

References

(1) Wiener, M.; White, S. Fluid bilayer structure determination by the combined use of x-ray and neutron diffraction. I. Fluid bilayer models and the limits of resolution. *Biophysical Journal* **1991**, *59*, 162–173.

(2) Wiener, M.; White, S. Fluid bilayer structure determination by the combined use of x-ray and neutron diffraction. II. "Composition-space" refinement method. *Biophysical Journal* **1991**, *59*, 174–185.

(3) Kučerka, N.; Nagle, J. F.; Sachs, J. N.; Feller, S. E.; Pencer, J.; Jackson, A.; Katsaras, J. Lipid Bilayer Structure Determined by the Simultaneous Analysis of Neutron and X-Ray Scattering Data. *Biophysical Journal* **2008**, *95*, 2356–2367.

(4) Kučerka, N.; Nieh, M.-P.; Katsaras, J. Fluid phase lipid areas and bilayer thicknesses of commonly used phosphatidylcholines as a function of temperature. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **2011**, *1808*, 2761–2771.

(5) Kučerka, N.; van Oosten, B.; Pan, J.; Heberle, F. A.; Harroun, T. A.; Katsaras, J. Molecular Structures of Fluid Phosphatidylethanolamine Bilayers Obtained from Simulation-to-Experiment Comparisons and Experimental Scattering Density Profiles. *The Journal of Physical Chemistry B* **2015**, *119*, 1947–1956.

(6) Pan, J.; Heberle, F. A.; Tristram-Nagle, S.; Szymanski, M.; Koepfinger, M.; Katsaras, J.; Kučerka, N. Molecular structures of fluid phase phosphatidylglycerol bilayers as determined by small angle neutron and X-ray scattering. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **2012**, *1818*, 2135–2148.

(7) Pan, J.; Marquardt, D.; Heberle, F. A.; Kučerka, N.; Katsaras, J. Revisiting the bilayer structures of fluid phase phosphatidylglycerol lipids: Accounting for exchangeable hydrogens. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **2014**, *1838*, 2966–2969.

(8) Pan, J.; Cheng, X.; Monticelli, L.; Heberle, F. A.; Kučerka, N.; Tieleman, D. P.; Katsaras, J. The molecular structure of a phosphatidylserine bilayer determined by scattering and molecular dynamics simulations. *Soft Matter* **2014**, *10*, 3716.

(9) Pan, J.; Cheng, X.; Sharp, M.; Ho, C.-S.; Khadka, N.; Katsaras, J. Structural and mechanical properties of cardiolipin lipid bilayers determined using neutron spin echo, small angle neutron and X-ray scattering, and molecular dynamics simulations. *Soft Matter* **2015**, *11*, 130–138.

(10) Marquardt, D.; Heberle, F. A.; Pan, J.; Cheng, X.; Pabst, G.; Harroun, T. A.; Kučerka, N.; Katsaras, J. The structures of polyunsaturated lipid bilayers by joint refinement of neutron and X-ray scattering data. *Chemistry and Physics of Lipids* **2020**, *229*, 104892.

(11) Doktorova, M.; Kucerka, N.; Kinnun, J. J.; Pan, J.; Marquardt, D.; Scott, H. L.; Venable, R. M.; Pastor, R. W.; Wassall, S. R.; Katsaras, J.; Heberle, F. A. The Molecular Structure of Sphingomyelin in Fluid Phase Bilayers Determined by the Joint Analysis of Small-Angle X-ray and Neutron Scattering Data. *The Journal of Physical Chemistry B* **2020**, acs.jpcb.0c03389.

(12) Marquardt, D.; Heberle, F. A.; Miti, T.; Eicher, B.; London, E.; Katsaras, J.; Pabst, G. 1H NMR Shows Slow Phospholipid Flip-Flop in Gel and Fluid Bilayers. *Langmuir : the ACS journal of surfaces and colloids* **2017**, *33*, 3731–3741.

(13) Rickeard, B. W.; Nguyen, M. H. L.; DiPasquale, M.; Yip, C. G.; Baker, H.; Heberle, F. A.; Zuo, X.; Kelley, E. G.; Nagao, M.; Marquardt, D. Transverse lipid organization dictates bending fluctuations in model plasma membranes. *Nanoscale* **2020**, *12*, 1438–1447.

(14) Kučerka, N.; Nagle, J. F.; Feller, S. E.; Balgavý, P. Models to analyze small-angle neutron scattering from unilamellar lipid vesicles. *Physical Review E* **2004**, *69*, 051903.

(15) Greenwood, A. I.; Tristram-Nagle, S.; Nagle, J. F. Partial molecular volumes of lipids and cholesterol. *Chemistry and physics of lipids* **2006**, *143*, 1–10.

(16) DiPasquale, M.; Nguyen, M. H. L.; Castillo, S. R.; Heberle, F. A.; Marquardt, D. *Methods in Membrne Lipids*; 2021; p TBA.

(17) Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* **2007**, *9*, 90–95.

(18) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362.

(19) Newville, M.; Stensitzki, T.; Allen, D. B.; Ingargiola, A. LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python. 2014; `https://doi.org/10.5281/zenodo.11813`.

(20) Doktorova, M.; Harries, D.; Khelashvili, G. Determination of bending rigidity and tilt modulus of lipid membranes from real-space fluctuation analysis of molecular dynamics simulations. *Physical Chemistry Chemical Physics* **2017**, *19*, 16806–16818.

(21) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A web-based graphical user interface for CHARMM. *Journal of Computational Chemistry* **2008**, *29*, 1859–1865.

(22) Brooks, B. R. et al. CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry* **2009**, *30*, 1545–1614.

(23) Wu, E. L.; Cheng, X.; Jo, S.; Rui, H.; Song, K. C.; Dávila-Contreras, E. M.; Qi, Y.; Lee, J.; Monje-Galvan, V.; Venable, R. M.; Klauda, J. B.; Im, W. CHARMM-GUI Membrane Builder toward realistic biological membrane simulations. *Journal of computational chemistry* **2014**, *35*, 1997–2004.

(24) Lee, J. et al. CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field. *Journal of Chemical Theory and Computation* **2016**, *12*, 405–413.

(25) Lee, J. et al. CHARMM-GUI Membrane Builder for Complex Biological Membrane Simulations with Glycolipids and Lipoglycans. *Journal of chemical theory and computation* **2019**, *15*, 775–786.

(26) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry* **2005**, *26*, 1781–1802.

(27) Phillips, J. C. et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *The Journal of Chemical Physics* **2020**, *153*, 044130.

(28) Klauda, J. B.; Venable, R. M.; Freites, J. A.; O'Connor, J. W.; Tobias, D. J.; Mondragon-Ramirez, C.; Vorobyov, I.; MacKerell, A. D.; Pastor, R. W. Update of the CHARMM All-Atom Additive Force Field for Lipids: Validation on Six Lipid Types. *The Journal of Physical Chemistry B* **2010**, *114*, 7830–7843.

(29) Klauda, J. B.; Monje, V.; Kim, T.; Im, W. Improving the CHARMM Force Field for Polyunsaturated Fatty Acid Chains. *The Journal of Physical Chemistry B* **2012**, *116*, 9424–9431.

(30) Giorgino, T. Computing 1-D atomic densities in macromolecular simulations: The density profile tool for VMD. *Computer Physics Communications* **2014**, *185*, 317–322.

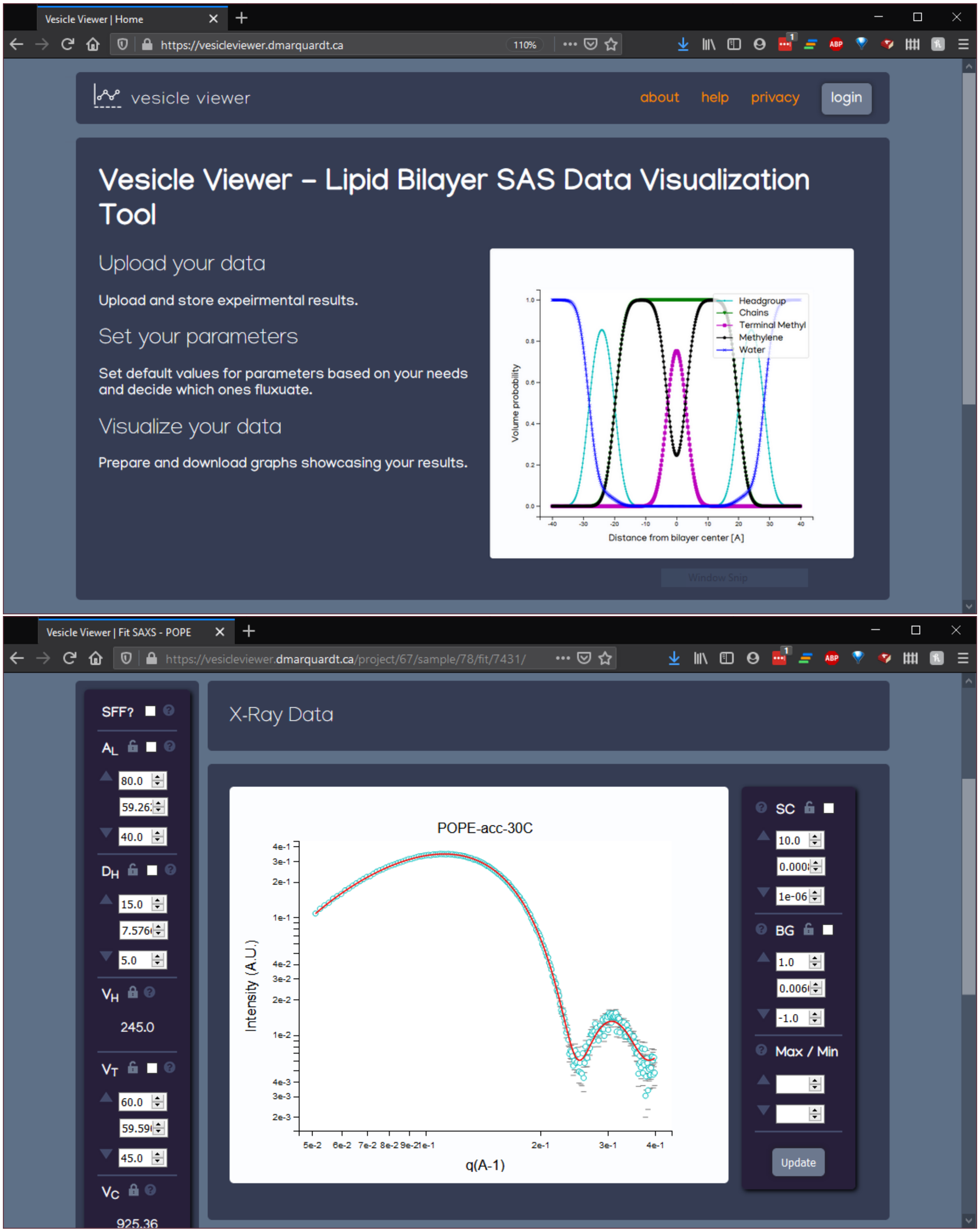
(31) Doktorova, M.; Heberle, F. A.; Marquardt, D.; Rusinova, R.; Sanford, R. L.; Peyear, T. A.; Katsaras, J.; Feigenson, G. W.; Weinstein, H.; Andersen, O. S. Gramicidin Increases Lipid Flip-Flop in Symmetric and Asymmetric Lipid Vesicles. *Biophysical Journal* **2019**, *116*, 860–873.

(32) Heberle, F. A.; Marquardt, D.; Doktorova, M.; Geier, B.; Standaert, R. F.; Heftberger, P.; Kollmitzer, B.; Nickels, J. D.; Dick, R. A.; Feigenson, G. W.; Katsaras, J.; London, E.; Pabst, G. Subnanometer Structure of an Asymmetric Model Membrane: Interleaflet Coupling Influences Domain Properties. *Langmuir* **2016**, *32*, 5195–5200.

(33) Nguyen, M. H. L.; DiPasquale, M.; Rickeard, B. W.; Doktorova, M.; Heberle, F. A.; Scott, H. L.; Barrera, F. N.; Taylor, G.; Collier, C. P.; Stanley, C. B.; Katsaras, J.; Marquardt, D. Peptide-Induced Lipid Flip-Flop in Asymmetric Liposomes Measured by Small Angle Neutron Scattering. *Langmuir* **2019**, *35*, 11735–11744, PMID: 31408345.

(34) Doktorova, M.; Heberle, F. A.; Eicher, B.; Standaert, R. F.; Katsaras, J.; London, E.; Pabst, G.; Marquardt, D. Preparation of asymmetric phospholipid vesicles for use as cell membrane models. *Nature Protocols* **2018**, *13*, 2086–2101.

(35) Eicher, B.; Heberle, F. A.; Marquardt, D.; Rechberger, G. N.; Katsaras, J.; Pabst, G. Joint small-angle X-ray and neutron scattering data analysis of asymmetric lipid vesicles. *Journal of Applied Crystallography* **2017**, *50*, 419–429.



Vesicle Viewer | Home

https://vesicleviewer.dmarquardt.ca

vesicle viewer

about help privacy login

Vesicle Viewer – Lipid Bilayer SAS Data Visualization Tool

Upload your data

Upload and store experimental results.

Set your parameters

Set default values for parameters based on your needs and decide which ones fluxuate.

Visualize your data

Prepare and download graphs showcasing your results.

Vesicle Viewer | Fit SAXS - POPE

https://vesicleviewer.dmarquardt.ca/project/67/sample/78/fit/7431/

SFF? □

A_L □

80.0

59.26:

40.0

D_H □

15.0

7.576

5.0

V_H

245.0

V_T □

60.0

59.59

45.0

V_C

925.36

X-Ray Data

SC □

10.0

0.000:

1e-06

BG □

1.0

0.006

-1.0

Max / Min

Update

Figure 1: Screenshots of (top) the main welcome page and (bottom) the fit page on the Vesicle Viewer website. The fit page showcases the parameter selection panel (left), fit graph including data points and the fit line (middle) and the scale and background adjustment panel (right).

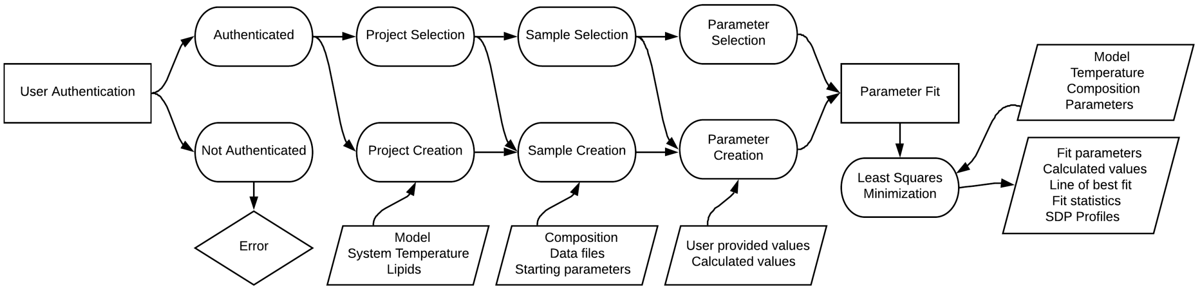


Figure 2: A flowchart describing the path taken as the user moves through Vesicle Viewer. The process begins with user authentication, and proceeds as various inputs are requested. All data collected from a user is connected to their account and visible only to them. Users may move forward or backward through the program as they like - data is selectively displayed to them to help guide them through the process and keep things organized.

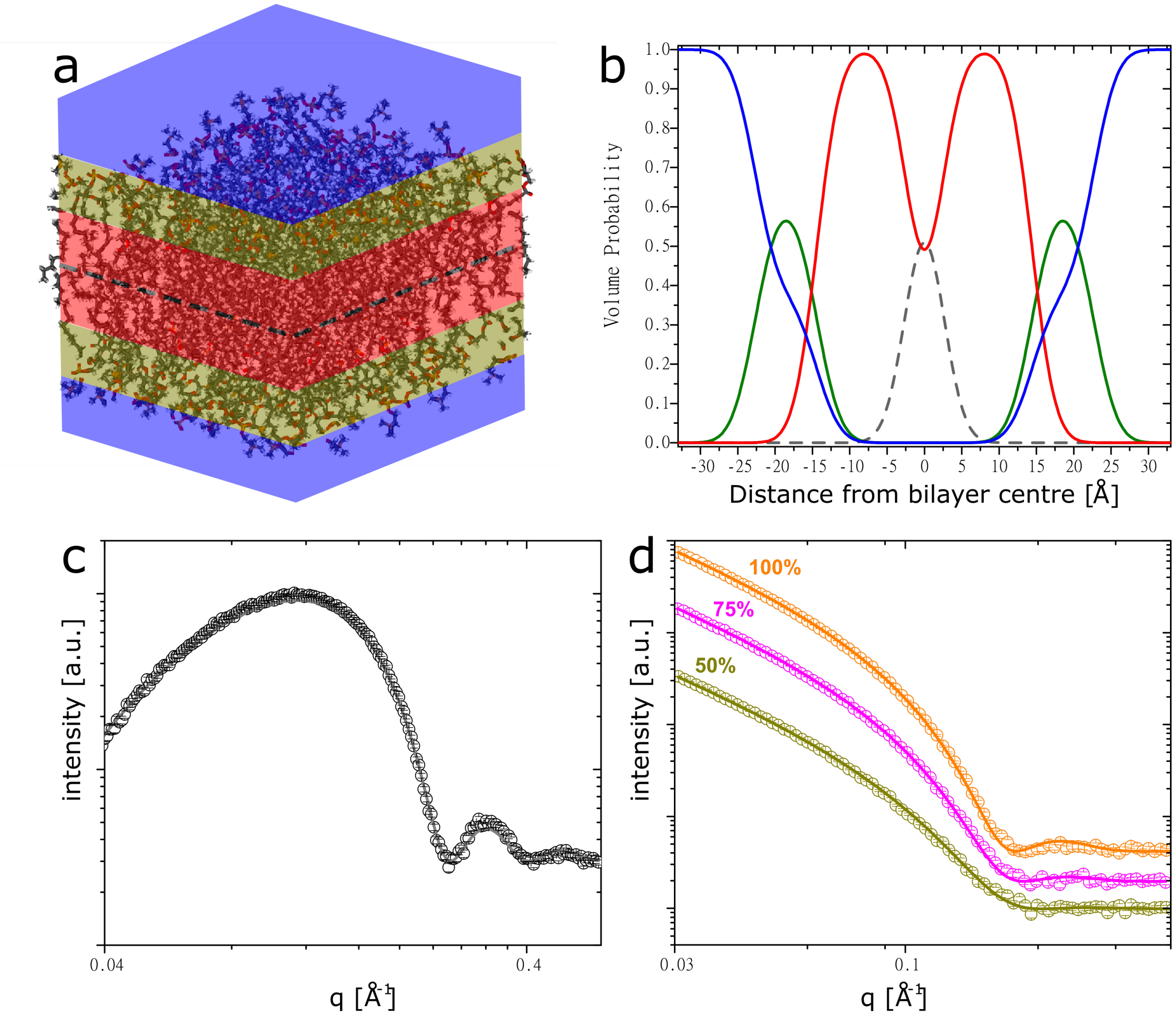


Figure 3: **a)** Schematic of POPC bilayer indicating the EZ-SDP slabs. Slab colours correspond to the water (blue), headgroups (gold) the methylene (red) and the terminal methyl at the midplane (dashed line). These colours correspond to the volume probability distributions of **b**. **c** and **d** are the SAXS and SANS respectively. Open symbols indicate data generated from the MD simulations and the solid lines represent the VV optimized models.The D_2O contrasts are indicated as 100%, 75% and 50% D_2O

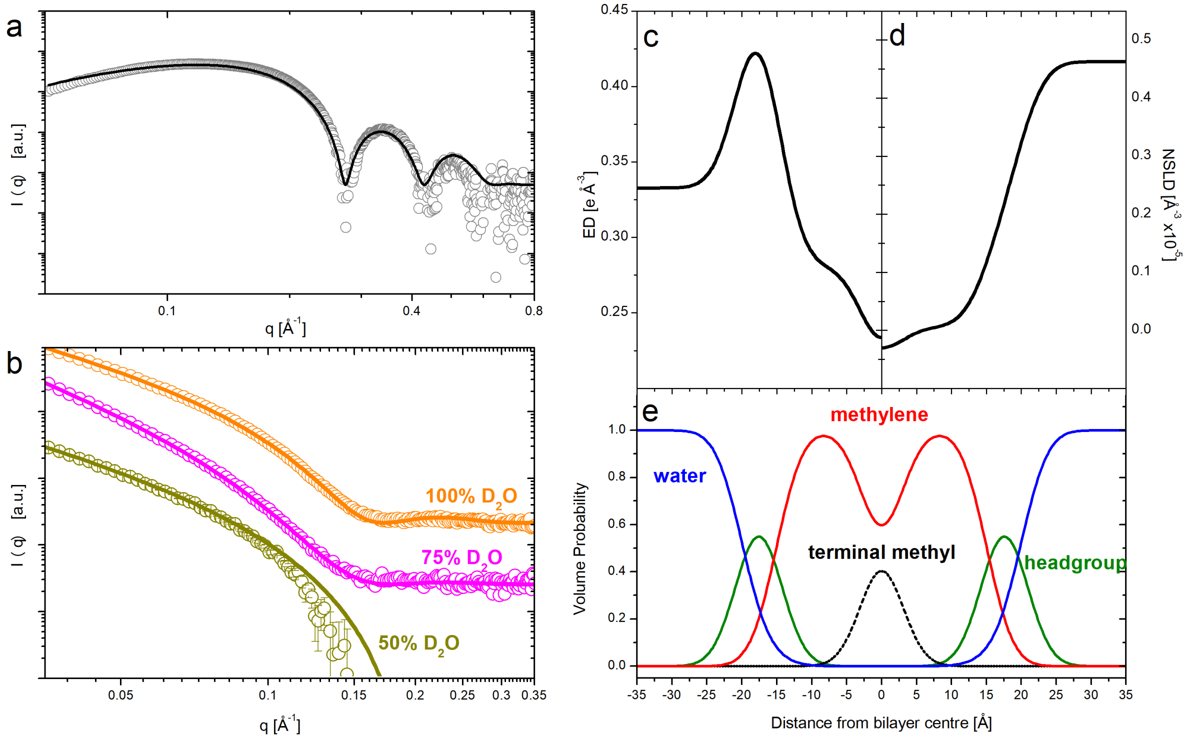


Figure 4: **(a)** and **(b)** are the SAXS and SANS curves, respectively. The open symbols are experimental data and the solid line represent the optimized model. Note, the 50% D_2O SANS curve of panel **(b)** appears cut-off due due to the increased incoherent background and the requirement that a logarithmic scale must be ≥0. **(c)** and **(d)** are the electron density (ED) profile and the neutron scattering length density profile from the optimized model, respectively.