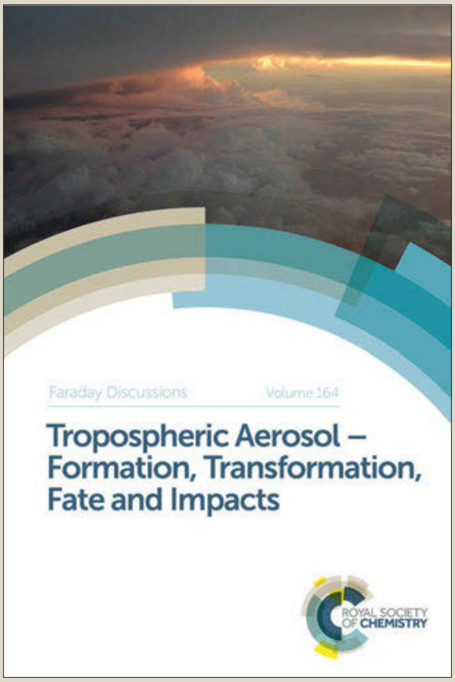

Faraday Discussions
Volume 164
Tropospheric Aerosol – Formation, Transformation, Fate and Impacts
ROYAL SOCIETY OF CHEMISTRY

Methodologies for the Analysis of Instantaneous Lipid Diffusion in MD Simulations of Large Membrane Systems

Matthieu Chavent[1,†,*], Tyler Reddy[1,†], Joseph Goose[1], Anna Caroline E. Dahl[1], John E. Stone[2], Bruno Jobard[3] and Mark S.P. Sansom[1]

Received XXX, Accepted XXX

DOI: 10.1039/b000000x

Interactions between lipids and membrane proteins play a key role in determining the nanoscale dynamic and structural properties of biological membranes. Molecular dynamics (MD) simulations provide a valuable tool for studying membrane models, complementing experimental approaches. It is now possible to simulate large membrane systems, such as simplified models of bacterial and viral envelope membranes. Consequently, there is a pressing need to develop tools to visualize and quantify the dynamics of these immense systems, which typically are comprised of millions of particles. To tackle this issue, we have developed visual and quantitative analyses of molecular positions and their velocity field using path line, vector field and streamline techniques. This allows us to highlight large, transient flow-like movements of lipids and to better understand crowding within the lipid bilayer. The current study focuses on visualization and analysis of lipid dynamics. However, the methods are flexible and can be readily applied to e.g. proteins and nanoparticles within large complex membranes. The protocols developed here are readily accessible both as a plugin for the molecular visualization program VMD and as a module for the MDAnalysis library.

1 Introduction

Our concept of biological membranes has evolved over the last 40 years, from the "fluid mosaic model"[1] to a more nuanced view of a laterally heterogeneous patchwork of proteins and lipids[2,3]. Membrane protein-lipid interactions are an area of intense research[4,5] with a number of membrane characteristics currently under scrutiny, including lateral diffusion of lipids, formation of nano-domains, and local membrane curvature[6,7]. The nature of lipid diffusion within membranes remains a topic of active discussion, with two major models being proposed:

- The *rattle in a cage* diffusion model describes short-range lipid movement: a lipid jumps from one location to another when it is able to escape from its immediate neighbours; and
- The *flow-like* diffusion model describes lipid movement at longer ranges: lipids can form loosely-packed clusters that move in concert, analogous to the movement of current.

[1] *Department of Biochemistry, University of Oxford, South Parks Road, Oxford OX1 3QU, United Kingdom* [*] *E-mail: matthieu.chavent@bioch.ox.ac.uk*

[2] *Beckman Institute University of Illinois 405 N. Mathews Avenue Urbana, IL 61801, USA*

[3] *LIUPPA laboratory, University of Pau, 1, av. de l'Université, 64013 Pau Cedex, FRANCE*

[†] *These authors contributed equally to this work*

While it remains difficult to experimentally assess the former theory, recent computational[8,9] and experimental[10] results tend to support the flow-like diffusion model. There has been a gap between the properties observable via computational approaches compared to those accessible via experiments[11]. However, this gap is now closing as increasing computational power provides access to simulations of biologically relevant times and length scales. To model large membrane systems, it is possible to use different levels of granularity in molecular dynamics (MD) simulations, ranging from all-atom[12] (AT) simulations to coarse-grained[13] (CG) models in which small numbers of atoms are grouped into representative particles, to Dissipative Particle Dynamics (DPD) simulations in which lipids may be modelled by just three[14] to eight particles[15]. As a result, the size and the complexity of computationally-accessible membrane systems are also increasing[16]. The membrane simulation field has thus progressed from e.g. a single α-helix embedded in a small patch of membrane for a duration of 1-2 ns[17] to the modelling of e.g. photosynthetic membrane vesicles[18] and HIV virion membranes[19,20]. At this level, visual analysis becomes challenging[21,22]. Despite using the latest graphical hardware for efficient system display[23-26], interpretation of very large membrane simulations by visual inspection is becoming unproductive due to their intrinsic complexity. There is therefore a substantial motivation to provide a simple and visually clear methodology to understand the complex dynamics of large-scale membrane systems. The challenge of computational visualization of very large systems has already been addressed in a number of areas of physics[27,28]. To some extent, visualization of very complex biomembrane systems, especially in the context of results from MD simulations, is less well developed. For example, the display of lipids is often limited to the representation of static states (i.e. simulation snapshots) and does not effectively render the complex dynamics of such systems. Progress has been made e.g. using arrows to depict discretised movements[8] but approaches are needed to more fully capture complex dynamical processes.

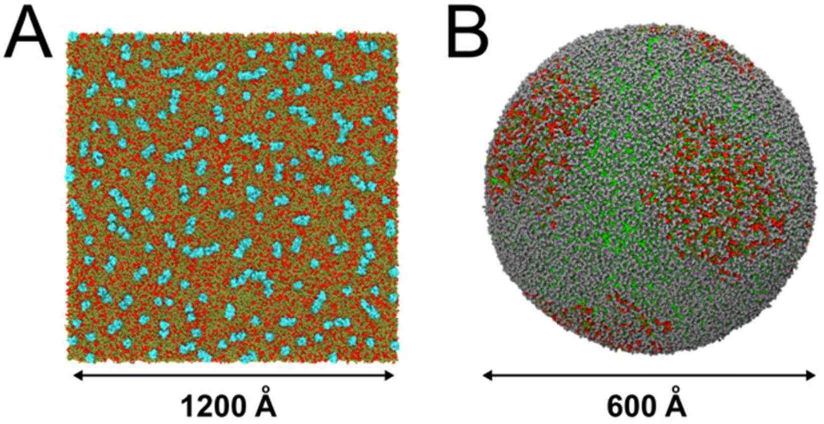


Fig. 1: Membrane simulation systems discussed in this study. (A) A planar membrane containing 256 OmpA proteins (cyan) in a bilayer of 28,260 molecules of POPE (brown) plus 9,420 molecules of POPG (red). (B) A spherical lipid vesicle comprised on 45,349 lipids (see Table 1 for details) including sphingolipids (grey), zwitterionic (brown) and anionic (red) phospholipids and cholesterol (green).

Here, we describe methods for informative visualization of lipid motion in large and complex systems using approaches derived from biomolecular simulations, physics, and computer visualization. We have developed visual and quantitative analysis methodologies for both molecular positions and their velocity field using vector field and streamline techniques. This allows us to highlight substantial and transient flow-like movements of lipids and to potentially assess the effects of crowding within models of bacterial cell membranes[29]. We have evaluated and applied our methodology using two examples of large biomembrane systems recently simulated in our research group: First, a model of a crowded, planar bacterial membrane with lateral dimensions ~1200 Å and containing 256 copies of a simple bacterial outer membrane protein (OmpA); second, a 300 Å radius vesicle the lipid composition of which matches the influenza virion membrane as determined in recent lipidomics studies[30] (see **Fig. 1**). With membrane surfaces of more than 10,000 Å^2 (~14,400 Å^2 and ~11,300 Å^2 for the planar membrane and the vesicle respectively), these systems are representative of the largest currently simulated by CG approaches[31-33]. Furthermore, these systems provide two distinct test cases, namely a planar membrane with a simple lipid composition but with crowded membrane proteins compared with a vesicle system containing a more complex lipid mixture with a high concentration of cholesterol (see Methodology for full details). This allows us to test the robustness of our approach with rather different biomembrane model systems.

2 Methods

2.1 CG MD simulations

We have used two systems that differ in terms of shape and composition: a planar lipid bilayer with embedded OmpA proteins that mimics a crowded bacterial membrane, and a vesicle mimicking the cholesterol-rich lipid bilayer component of an influenza virion membrane but without any embedded proteins. Due to these intrinsic differences, the two systems are expected to exhibit distinct dynamic behaviour. Hence, they provide realistic test cases to evaluate our approach.

2.1.1 Planar membrane

This model mimics a large patch of a crowded bacterial membrane. The initial system size was 1180 x 1180 x 104 Å^3 and contained 37,680 lipids (1:3 POPG to POPE) and 256 OmpA proteins, giving a protein:lipid ratio of roughly 1:150 (see **Table 1**). The total number of particles is ~1,202,500 (~577,900 excluding solvent). The system was modelled using GROMACS 4.5.3 in combination with a modified in-house version of the MARTINI forcefield[34,35] at a temperature of 313K. Bilayers of this size cannot be readily self-assembled directly due to system instability caused by pressure coupling and initial fluctuations. Instead we self-assembled and equilibrated a smaller patch containing approximately 150 lipids which was then replicated. POPE and POPG do not phase separate and are randomly distributed within the bilayer during this initial equilibration. The small lipid patch was then analyzed for leaflet asymmetry and corrected such that the numbers of lipids in the two leaflets were balanced. The patch was then tiled to form a 4x4 patch containing ~2,500 lipids followed by a second equilibration and another round of tiling, resulting in a final patch containing ~38,000 lipids. The final equilibration was performed for 100 ns. During each round of equilibration after membrane tiling, the system was observed to shrink modestly in the x-y plane and expand slightly in the z plane. The final system was simulated for 1 µs, during which frames were saved every 0.2 ns giving a 5000-frame trajectory for subsequent analysis.

2.1.2 Vesicle

In the course of constructing a full-scale computational model of a human influenza A virion, a vesicle with lipid composition matching the experimentally-determined influenza A lipidome[30] was produced. The details of this construction process will be described in more detail elsewhere (Reddy et al., ms. in preparation). In brief, the vesicle construction process starts off with random placement of lipids within two leaflets of a spherical system with an outer diameter of ~740 Å. The random seeding of lipid positions within a set of spherical geometric constraints was handled by employing the Packmol program[36] for addition of lipids interleaved with energy-minimization in GROMACS. The asymmetric lipid distribution in the vesicle was based on the data of Gerl et al.[30]. Thus the inner leaflet contains POPS, DOPX (ether-linked DOPE) and DOPE with an approximate 3:2:1 ratio, while the outer leaflet contains sphingomyelin (SM) molecules (with saturated C16 chains). Cholesterol (CHOL) molecules were distributed between the two leaflets and comprise >50% of all lipid molecules in the vesicle. Some mixing (*i.e.,* flip-flop) of leaflet components occurred during the equilibration process. An inner core of restrained CG particles was employed to mimic the interior of the virion which was not modelled explicitly. The system contained ~3,006,000 CG particles in total (~500,000 without solvent). Lipids were initially somewhat sparsely packed, forming a well-packed lipid vesicle (~600 Å outer diameter) after a 300 ns equilibration in water at 323 K using GROMACS 4.5.5[37] and the MARTINI 2.1 forcefield[38,39]. The resultant lipid vesicle model was simulated for 5 µs at 295K, with the final microsecond being analysed in the current study.

2.2 System pre-processing

It is necessary to perform a series of pre-processing operations on the MD simulation data prior to employing our analysis methodologies. First, for each system, the motion of the centre of mass was removed to avoid possible artefacts. Subsequently, we performed low-pass filtering to remove high frequency noise using the GROMACS g_filter function. The filter shape is $\cos\left(\frac{\pi t}{N}\right) + 1$ between -*N* and +*N,* where N is the number of frames, as detailed in Affentranger *et al*.[40]. In order to study the effect of such filtering on visual results, we have filtered the two systems over a series of time scales: 10 frames, 20 frames and 40 frames corresponding respectively to 2 ns, 4 ns and 8 ns. The largest window used corresponds to 0.8% of the simulation. To study the behaviour of each lipid membrane leaflet independently they were separated using in-house python code with the MDAnalysis library[41].

2.3 Diffusion analysis and clustering

The mean square deviation (MSD) was calculated using the g_msd module in the GROMACS 4.5.4 package[37,42]. This module was previously employed to calculate lipid diffusion coefficient[43,44] using the Einstein relation. “Restart” parameters of 2 ns and 5 ns were compared to measure their influence on the MSD curve and diffusion coefficient. Protein clustering was quantified using an in-house tcl script. To do so, the distance between the centre of mass of each protein was calculated and if this distance was smaller than the radius of the protein plus 5 Å we assumed that the two proteins were in contact. Each protein neighbour at time *t* was stored and the clusters were identified by a connectivity test. The calculations were performed every 10 ns on the 1 µs trajectories.

2.4 Algorithm implementation and availability

The algorithms presented here are primarily implemented in both Tcl and Python. The Tcl code will be incorporated as a plugin for the molecular visualization program VMD[45] while the Python code is designed to be a module for the MDAnalysis library[41]. Scripts as well as documentation are available at the address: http://sbcb.bioch.ox.ac.uk/flows/

2.4.1 Path line visualization

The path line visualization is currently only implemented in VMD. We have used VMD 1.9 to display and calculate the performance of such rendering. In the case of the planar membrane we used a tail of 5 frames (1 ns) and a mean displacement value of 1.2 Å. To obtain an equivalent rendering for the vesicle system, we chose a window size of 20 frames (4 ns) and a mean displacement value of 0.5 Å.

2.4.2 Vector field visualization

The vector field visualization is executable from within either VMD or a Python module depending on MDAnalysis. For VMD, it is based on Tcl code launchable from the VMD plugin. For this representation, two pieces of code are available: a 2D version and a 3D version. The 2D version flattens all the z values on one layer while the 3D version use the complete 3D coordinates. Plotting in the MDAnalysis module (written in Python) leverages the matplotlib library in 2D[46] and the MayaVi[47] package in 3D.

2.4.3 Streamline visualization

Streamline visualization is executable from within either VMD or a Python module depending on MDAnalysis. Our in-house Python code used the matplotlib[46] streamplot[i] function to produce 2D streamlines and, for the 3D streamlines visualization, we used the MayaVi package[47].

3 Results and Discussion

In parts 3.2 to 3.4 we focus on the movement of lipid phosphate headgroup particles for one leaflet to demonstrate our method on a simple case. However, the method can be applied to other cases like water molecules diffusion.

3.1 Global analysis of dynamic behaviour of lipids and proteins

Prior to applying our flow analysis tools, we will briefly describe the dynamic and clustering behaviour of lipids and proteins in our simulation systems.

3.1.1 Clustering of proteins

The planar membrane model (1:150 protein-lipid ratio or ca. 15% of the surface occupied by proteins) provides an example of a relatively crowded membrane system (see **Fig. 2**). Our final snapshot at 1 μs depicts a crowded membrane environment with numerous clusters of OmpA proteins, qualitatively consistent with recent results from high-speed AFM[48] studies of protein clustering in bacterial outer membranes. Visual inspection reveals approximately linear clusters within the bilayer plane. Comparable *chains* of interacting proteins were recently highlighted by computational work for smaller systems[29]. At the end of the simulation, the system consists of 24 single proteins, 27 dimers, 25 trimers and 9 tetramers (see **Fig. 2-C**). There are also 11 clusters formed by more than four proteins. The largest cluster (of 9 proteins) is highlighted in **Fig. 2-A**.

[i] http://matplotlib.org/examples/images_contours_and_fields/streamplot_demo_features.html

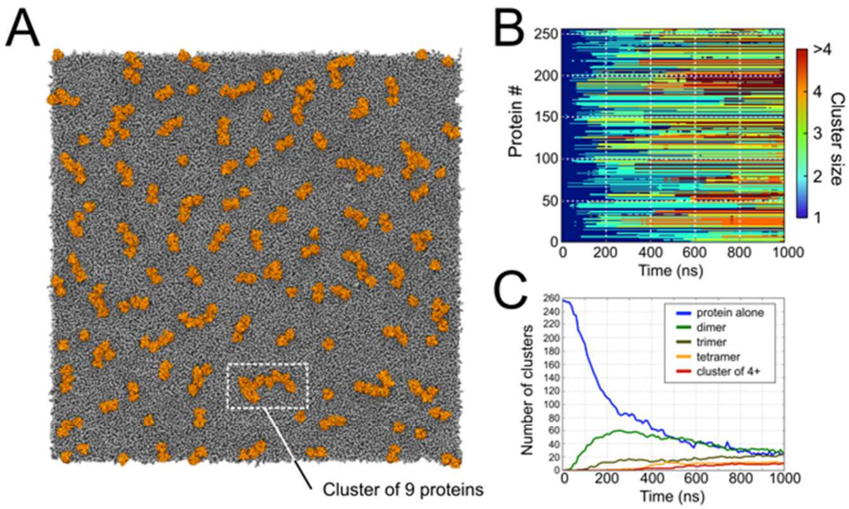


Fig. 2: Protein clustering analysis for the planar bacterial membrane model. (A) VMD snapshot of the planar membrane at the end of the 1 µs, with the proteins shown in orange. (B) Aggregation of proteins as a function of time. Each line represents a unique protein and the colours illustrate each protein's association to a cluster of a certain size. A cluster of size one corresponds to a single, unclustered protein. (C) The frequency of occurrence of clusters of different sizes as a function of time.

We also calculated the clustering of these proteins as a function of time (see **Fig. 2 B, C**). Analysis reveals that the number of monomeric OmpA proteins decreased rapidly (also see **Sup. Movie 1**), with conversion of monomers to dimers reaching a maximum after ~250 ns. Dimers recruit additional monomeric proteins to form trimers, and may also aggregate to form larger clusters. After 200 ns, we observed the formation of tetramers followed by the formation of higher order clusters at ~300 ns. The clusters continued to evolve reaching a nearly steady state by the end of the simulation.

3.1.2 Diffusion of lipids and proteins

We have estimated the diffusion coefficients for proteins and for lipids in the two systems (see **Table 1**). As anticipated, the OmpA proteins diffuse more slowly than lipids in the planar membrane, with the two lipid species, POPE and POPG, showing similar diffusion characteristics. We have also calculated the diffusion coefficient of these lipids in each leaflet separately and the results were equivalent to the ones in Table 1. The diffusion coefficient for OmpA proteins is in agreement with the value of ~1.5 x 10^{-7} $cm^2.s^{-1}$ obtained recently for a matching system composition of smaller membrane size[29].

The vesicular lipid diffusion coefficients are clearly lower by a factor of 20 compared to the planar membrane. Although this may be explained in part by the lower temperature (295K for vesicle; 313K for planar membrane) it is mainly due to the high content (> 50% molecular species) of cholesterol[49]. These diffusion coefficients are not far the value of 0.7 x 10^{-7} $cm^2.s^{-1}$ obtained using magic angle spinning NMR spectra experiments on an influenza lipid vesicle at 290 K[50]. If we compare between lipids, the differences in diffusion coefficients relate to the asymmetry in the vesicle. The inner leaflet is mainly composed of POPS, DOPE and DOPX lipids while the outer leaflet is largely comprised of SM, with CHOL present in both leaflets but at a higher level in the outer (62%) relative to the inner (38%) leaflet. So, the diffusion coefficients reveal an inner leaflet which is ca. 2-3 times more diffusionally mobile than the outer leaflet.

Table 1: Lateral diffusion coefficients for the membrane constituents of both systems. The values of D_{xy} were derived from the MSD curves shown in Supporting Information Fig. S1, by fitting these curves between 100 and 900 ns for each system. The error estimates are included in brackets.

Components	Number of molecules	D_{xy} (10^{-7} cm^2.s^{-1})
Planar membrane		
OmpA	256	2.04 (± 0.26)
POPE	28260	5.16 (± 0.14)
POPG	9420	5.14 (± 0.14)
Vesicle		
POPS	6802	0.23 (± 0.01)
DOPX	4081	0.24 (± 0.01)
DOPE	2268	0.24 (± 0.01)
CHOL	23804	0.17 (± 0.01)
SM	8389	0.08 (± 0.01)

Whilst the calculation of average diffusion coefficients can provide insights into the overall lateral fluidity of a membrane it does not reveal more complex spatial and temporal variations in behaviour for a given species in different regions of the membrane (e.g., proximal versus distal to proteins). Furthermore, analysis of e.g. lipid clustering is complex due to their loose, dynamic packing[8,9,51]. For these reasons, we have applied specialized visualization methodologies to our MD simulation systems to provide a more detailed insight into the dynamic properties of lipids in membranes. Thus the dynamic dissimilarities of the two membrane systems we have simulated provide test cases to assess the robustness of our approach.

3.2 Following each lipid by path line visualization

Current published methodologies to visualise the displacement of lipids fall in two categories:

1) Using arrows to represent individual lipid displacement between a time *t* and a time *t+dt* [8,9,52,53].
2) Using lines to connect the different positions of individual lipids over time[8,54,55].

The latter is generally used to present an overview of the global region explored by a lipid while the former is used to depict lateral displacements of lipids in a dedicated area between time *t* and *t+dt*.

We have combined these two representations to a single depiction that can evolve as a function of time to enhance visual interpretation of molecular mobility in a membrane. In practice, lines (or cylinders) depict the path of lipids over time like a tail. For a defined time window, positions of a lipid at different times (t_i through t_n) are linked to the current time, *t* **(**see **Fig. 3 A)**. At time *t+dt* the earliest element is removed while a new line is added at the front of the tail segment (see **Fig. 3 B**). We have coloured segments and set the thickness of the tail based on the displacement between two time steps (see **Fig. 3 C**). This highlights changes to lipid motion over time and removes the noise of quasi-static lipids.

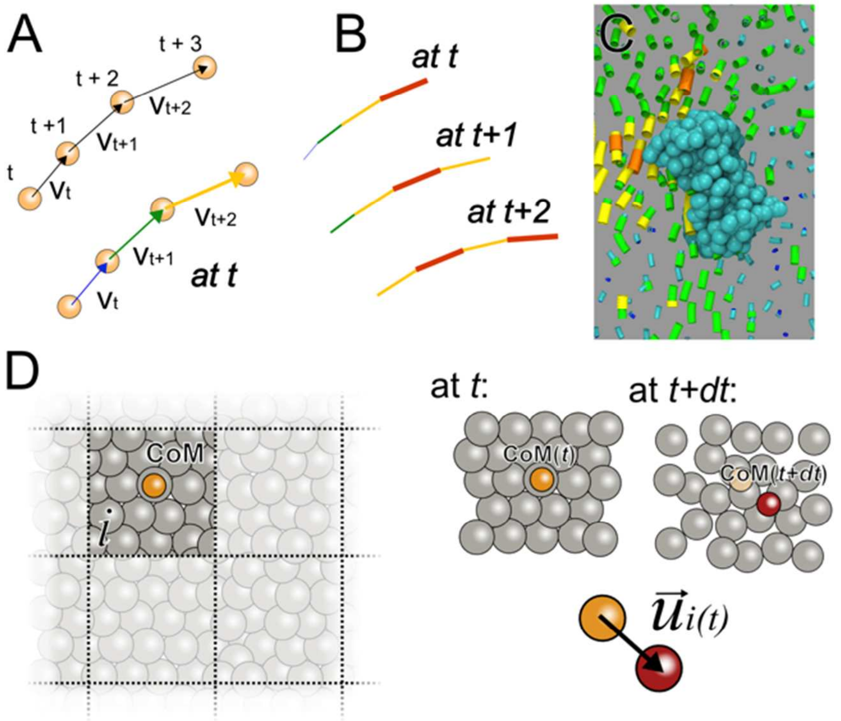


Fig. 3: Path lines and vector field construction. (A) Construction of path lines. (B) Evolution of a path line in function of the time. (C) Lipid path lines around proteins. For (B) and (C), colour and thickness of the tail are function of lipid displacement between 2 time steps. (D) Simple 2D schematic of the creation of the vector field. After defining a grid, the centre of mass of each lipid selection (in each grid square) is calculated at time *t*. Then, the centre of mass of the same selection is calculated at time *t+dt*, and the two centres are used to define the vector.

This first approach allows visualizing large and transient correlated lipid movements for the planar system (see **Fig. 4 A** and **Sup. Movie 1**). These transient movements can be linear as well as rotational - *i.e.* vortices. This behaviour has been reported in previous computational[56,57] and experimental[58] studies. Furthermore, **Sup. Movie 1** highlights transient correlation between the lipid movements and the displacements of proteins. The translational and rotational motions of the lipids appear to be correlated with comparable motions of the proteins. A more detailed analysis of such phenomena will be presented elsewhere (Goose, Chavent & Sansom, in preparation). In contrast, the vesicle system presents only small changes and did not exhibit large well-defined coherent lipid motions (**Fig. 4 B** and **Sup. Movie 2**). Instead, this system appeared to present large patches of motionally stable lipids.

In some respects, analysing the membranes in these systems may be likened to analysing macroscopic systems such as oceans or atmosphere in which transient flows appear[ii]. To study such flow formation, it is common to use evenly-spaced probes depicting the underlying vector field.

[ii] For a comparison see http://www.nasa.gov/topics/earth/features/perpetual-ocean.html or http://hint.fm/wind/

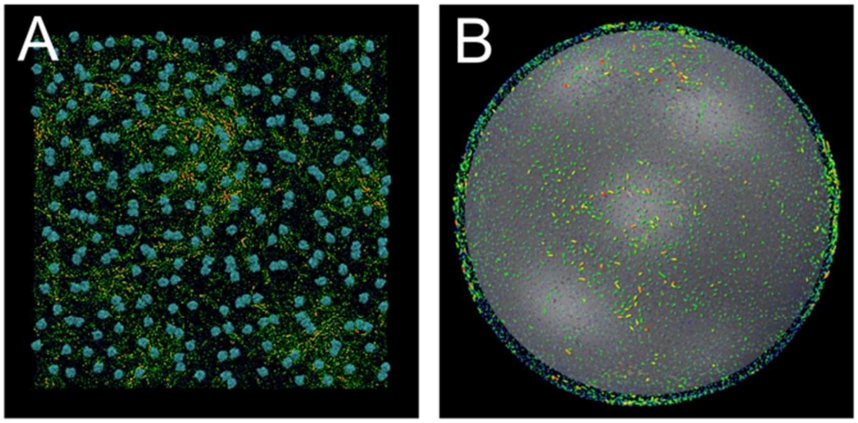


Fig. 4: Path line visualization of a filtered (8 ns time window) dataset. (A) Path line rendering of the upper leaflet of the large planar membranes containing 256 OmpA proteins. (B) Path line rendering of the outer leaflet of the large vesicle membrane. Movies of (A) and (B) are available in the Supplementary Material. Snapshots were captured with the tcl plugin in VMD.

3.3 Approaches for following a group of lipids

3.3.1 Vector field visualization

In material physics, vector field visualization has been used to detect phase transitions in large ferroelectric materials[27]. In computational biology, vector fields can be used to depict evolution of regulatory networks[59] or to study the role of water molecules in protein folding processes [60,61]. Here, we use a vector field approach to follow a group of lipids and thereby simplify the representation of their dynamics.

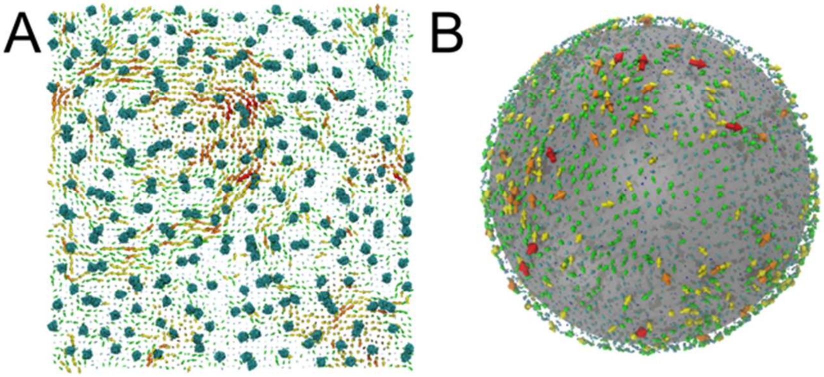


Fig. 5: Vector field visualization using a 20 Å grid resolution. (A) 2D results on the planar membrane. (B) 3D results on the vesicle system, for which a transparent sphere was added to clarify the visualization. Snapshots were captured with the tcl plugin in VMD.

Fig. 3 D presents the simple steps used to create a vector field:
1) The systems were divided into grids. For each cell *i* (square in 2D or cube in 3D), we calculate the centre of mass, $CoM_{i(t)}$, of the constituent lipids at a time *t*.
2) Then, we calculate $CoM_{i(t+dt)}$, the centre of mass of the same selection at time *t*+*dt*,
3) These two centres of mass are used to define the vector at time *t* following the equation:

$$\vec{u}_{i(t)} = CoM_{i(t+dt)} - CoM_{i(t)} \qquad (1)$$

The vector origin is the centre of mass of the lipids at time *t*. In this paper, we have chosen a *dt* = 0.2 ns (*i.e.* 1 frame). This value is quite small as we wish to avoid potentially large spread of the lipid selection at *t*+*dt*. For the examples presented below we have used a grid with a resolution of 20 Å.

This visualization has some advantages over the path line visualization. Reducing the number of graphical objects (*i.e.* vectors representing a group of lipids here instead of lines corresponding to each lipid in the previous representation) evidently clarify the representation (**Fig. 5** and **Sup. Fig. 2**). It is especially valuable for the creation of a static picture. Furthermore, this representation provides more insights because it is possible to track the direction of the lipid movements using arrows. This vector field can be then used as input to perform more advanced rendering.

3.3.2 Streamlines visualization

Mathematically, streamlines are curves instantaneously tangent to the motion vectors. Streamline curves have been used for several decades to represent flow patterns[62]. A streamline is a curve that describes the trajectory of a particle in a stationary vector field at a given time. More formally, given a 2D time-dependent vector field $\mathbf{v}(x,t) = (v_x,v_y)$, a streamline S is a parametric curve $S(\tau)$ defined at time t and initiated from a seed point p. $S(\tau)$ is given by the equation:

$$\frac{dS}{d\tau} = \mathbf{v}(S(\tau),t) \text{ with } S(0) = p$$

Any sample point of the streamline S is given by:

$$S(l) = S(0) + \int_0^l \mathbf{v}(S(\tau),t)d\tau$$

with the curvilinear coordinate $l \in [-l_1, l_2]$ and $l_1, l_2 \geq 0$. A standard Runge-Kutta integration scheme[iii] is used to sample a streamline backward ($l \in [-l_1, 0]$) and forward ($l \in [0, l_2]$) from its seed point. We can notice that in function of the algorithm used the streamlines can be integrated in one direction or in both directions. Then, the placement of the streamlines constitutes a key step and has been studied for a long time[63].

An analogous approach, display of field lines, is used in computational biology to depict electrostatic fields calculated with algorithms such as the Adaptive Poisson-Boltzman Solver (APBS)[64]. Streamlines were also recently used to describe water flows revealed by MD simulations of photosystem II[65].

It is possible to display streamlines in 2D or 3D. This streamline representation can combine the advantages of the two previous representations. Like the vector field, it follows a group of lipids but not all the lipids while, like the path lines, the streamlines help to better delineate the "flow" pattern of the lipids creating continuous lines between discretised vectors (**Fig. 6**).

[iii] http://en.wikipedia.org/wiki/Runge-Kutta_method

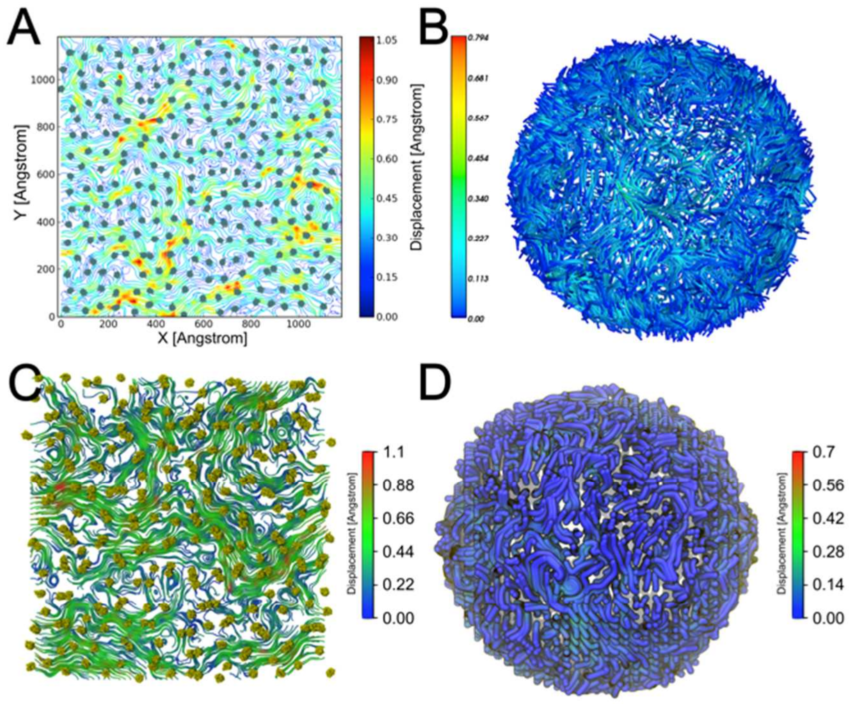


Fig. 6: Streamline visualizations for the planar membrane and the vesicle system. (A) and (C) 2D visualization of the upper leaflet of the planar membrane system. (B) and (D) 3D visualization of the vesicle outer leaflet. (A) and (B) are rendered using in-house Python code employing the MDAnalysis module and matplotlib/MayaVi while (C) and (D) are displayed using VMD program. For (A) and (D) associated movies are available in Supplementary Material.

For the planar membrane, this visualization enables display of both linear and circular nanometer-scale movements of lipids (**Fig. 7 A and B**) as well as small protein-proximal vortices (see **Fig. 7 C**). **Sup. Movie 3** shows that the correlated lipid patterns are transient (nanosecond scale lifetime) and constantly changing. Furthermore, visual inspection of the results tends to show that these flow patterns are correlated with areas of substantial lipid displacement. The streamlines rendering also highlight how the lipid flows can bridge several proteins far away from each other in large correlated motions (see **Fig. 6 A and C and Sup. Movie 3**) extending the model of protein and lipids dynamic complexes[66] to transient networks of protein and lipids.

The streamline visualization did not depict very large nanometer-scale flow for the vesicle system. It was only possible to display little vortices (see **Fig. 7 D** and **Sup. Movie 4**). This result is in agreement with the path line visualization and the calculation of the diffusion coefficient. It is also possible to use streamlines to compare correlated lipid motions between the two leaflets of a membrane (see **Fig. 7 D**).

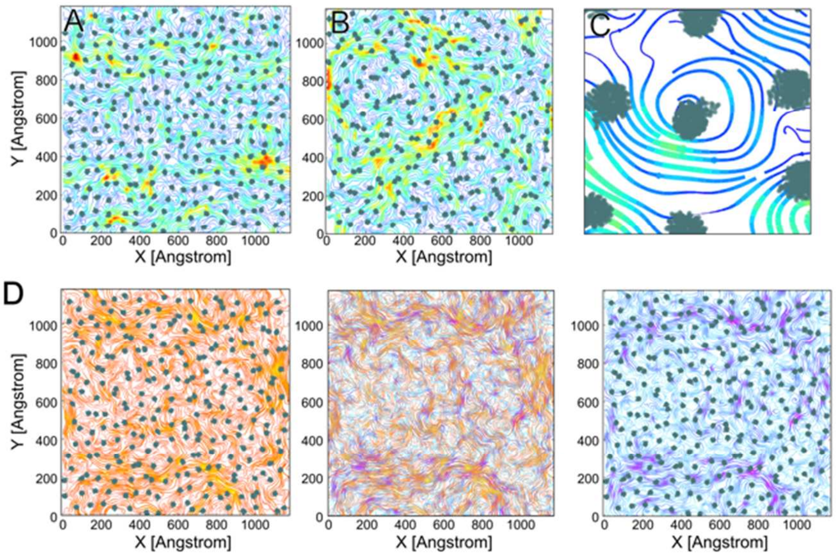


Fig. 7: Lipid motions patterns highlighted by the streamline visualization. (A) Large linear correlated motions crossing the entire system (for *y*~300 Å and *y*~900 Å). (B) Large circular correlated motions (from 0 to 800 Å in *x* and 400 to ~1100 Å in *y*). Same snapshot than Fig. 4 A and Fig. 5 A. (C) Vortex around a protein. (D) Comparison of upper and lower leaflets highlighting correlated areas across the bilayer. On left, upper leaflet; on right, lower leaflet; in the middle superimposition of the lower and the upper leaflet. Snapshots made with the MDAnalysis Python plugin.

3.5 Analysis tool to complement the visualizations: leaflet correlations

Streamline visualization revealed interesting flow lipid patterns. However, the representations may contain considerable information and it is therefore important to be able to isolate selected regions for comparisons. There have been several proposals to filter streamline data to aid such analysis (for examples see refs[67,68]). We applied comparable methodology to our systems to simplify the streamline representation using calculations based on the vector fields. As noted above, streamline visualizations can highlight correlated lipid motions between two leaflets. Here, we filter the data to highlight only the areas where the lipid motions are correlated above a certain threshold.

One can define $C_{l(t)}$, the leaflet correlation function at time *t*:

$$C_{l(t)} = \frac{1}{N}\sum_{i=1}^{N}\frac{\vec{u}_{i(t)}\cdot\vec{v}_{i(t)}}{|\vec{u}_{i(t)}||\vec{v}_{i(t)}|} \quad (2)$$

where · denotes the scalar product of vectors $\vec{u}_{i(t)}$ and $\vec{v}_{i(t)}$ as defined by equation (1). $\vec{u}_{i(t)}$ and $\vec{v}_{i(t)}$ are lower and upper leaflet vectors, respectively. *N* is the number of non-empty cells in the grid. This type of correlation was recently used to identify correlated intermolecular dynamics of short polymer chains[69]. Note that $-1 \leq C_{l(t)} \leq +1$, where -1 represents movements in opposite directions and +1 represents movements in the same direction. A $C_{l(t)}$ value near 0 depicts no correlation between the leaflets.

For the planar membrane we used the previously-defined Cartesian grid. Due to the shape of the vesicle, we used spherical coordinates (see Supplementary Material). To select the same areas of interest, we chose the same θ and φ between the two leaflets while changing the radial distance *r* to correspond to the average radius of phosphate headgroup particles in each leaflet. After obtaining these coordinates, we created cubes of dimensions 20 Å to select the lipids for vector calculations (as described in section 3.3).

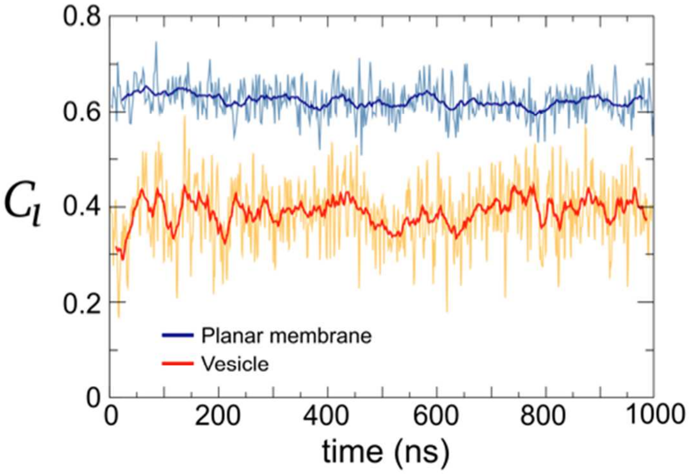


Fig. 8: Inter-leaflet lipid flow correlation assessment for the planar bilayer (blue) and the vesicle (red) systems. The bold lines depict the averaged values on 10 steps.

Fig. 8 presents the inter-leaflet lipid flow correlation results for each system using data filtered over an 8 ns time window. There is a high correlation between the two leaflets in the case of the planar membrane (with $C_{l(t)} > 0.6$) while, for the vesicle, the correlation values are lower, with $C_{l(t)} \sim 0.4$. These inter-leaflet correlations are rather more pronounced than those seen in simulations of a simple DOPC bilayer system[70]. The difference in the magnitude of the correlations between these two systems may be linked to the differences in their compositions. As noted above, the planar membrane has a symmetric, relatively simple lipid composition. In contrast the lipid composition for the vesicle system is highly asymmetric (see above), with a more slowly diffusing outer leaflet which is nearly twice as rich in cholesterol as the inner leaflet. We also note that the planar membrane correlation may be slightly overestimated due to the flattened 3D coordinates in the x-y plane.

The analysis of correlations may be used to filter the streamlines by using a cut-off threshold for these inter-leaflet lipid correlation values. For example, for the *E. coli* bilayer system, we can display the streamlines with a $C_{l(t)}$ value superior to 0.8 (see **Fig. 9**). This represents around 50% of vectors used to define the streamlines. This highlights large inter-leaflets patterns visually correlated with the areas of high velocity. We postulate that the slower diffusion of the proteins can be, in part, explained by the fact that sometime the flows between leaflets are not correlated which can have an influence on trans-membrane proteins. So, using this filtering can help to select only high correlated parts of the membrane for further analysis.

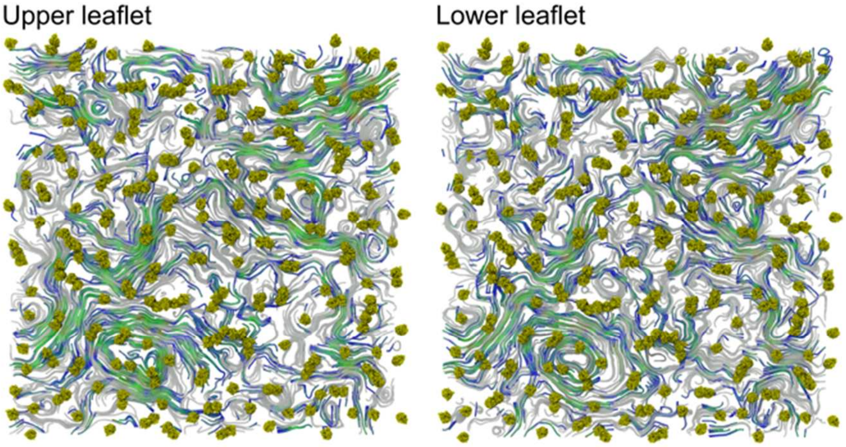


Fig. 9: Examples of correlated dynamics for the planar membrane visualized using VMD. Coloured streamlines depict the correlated parts between the two leaflets (with $C_{l(t)} > 0.8$) while the remaining streamlines are coloured in gray.

3.7 Low pass filtering

In this article we have used the filtered data with the largest sampling window (8 ns) for clarity, but it is important to note that the filtering granularity can influence the results. Correlated lipid motions have previously been highlighted for different time scales: over picoseconds[10,69], nanoseconds[8,9], and hundreds of nanoseconds[9]. We have therefore tried different filters to see if this can highlight a particular flow pattern on a specific time scale. Here, we examine the effect of the filter window size on the representation of the data.

We have used low pass filtering with the GROMACS g_filter function to filter the two systems over a series of time scales: 10 frames, 20 frames and 40 frames corresponding respectively to 2 ns, 4 ns and 8 ns. This approach can remove high frequency motions of lipids and potentially influence the results of this study. This type of method has previously been employed to lipid membrane-based MD simulation systems[71] and for polypeptide folding[40]. Recently, Gypsa and colleagues have also used a low pass filter to highlight different characteristics of membrane curvature[72].

Filtering has no effect on the diffusion coefficients calculated for our simulations, with any discrepancies between filtered and unfiltered values falling within the bounds of the estimated error (data not shown). We calculated the leaflet lipid motion correlation values for different filter window sizes (see **Sup. Fig. 3**). The results for the planar membrane indicate that the leaflet correlation is affected by the filtering. The raw data presents a leaflet correlation value greater than 0 ($C_{l(t)}$ ~ 0.3), consistent with some correlated lipid movement between leaflets in the absence of a filter. This value drastically increases with the addition of even a short time window filter ($C_{l(t)}$ ~ 0.55 for 2 ns filter). Increasing the time window size of the filter further does not lead to a substantial increase in calculated leaflet correlation, consistent with potential convergence to a consensus correlation value ($C_{l(t)}$ ~ 0.6 for 4 ns filter and $C_{l(t)}$ ~ 0.63 for 8 ns filter).

We reused equation (2) to compare the data obtained with different filter time window sizes within a single leaflet. To do this, instead of comparing vectors from 2 different leaflets, we compare vectors for the same leaflet but with different filter windows. We called this function filter correlation: $C_{f(t)}$. The raw data has a small correlation with 8 ns filtering ($C_{f(t)}$ ~ 0.1) but this value drastically increases even for a small filter over 2 ns ($C_{f(t)}$ ~ 0.5) and is even higher for data filtered on a 4 ns timescale ($C_{f(t)}$ ~ 0.75) (see **Sup. Fig. 4**). Indeed, visual inspection of the streamline plots indicates conserved patterns for all the filtered data (see **Sup. Fig. 5**). The effect of this filtering is to remove high frequency noise, which can potentially be related to jiggling motions of lipids around their positions. Thus, the filtering can allow focusing of the data representation over longer time scale motions rather than residual noise from local fluctuations.

3.8 Algorithm comparison and efficiency

We have tested our implementations on an Intel 6 cores (12 threads) *i7*-3930K with a Geforce GTX 680. The tcl code was used on 1 thread while the Python code, implemented to handle multithreads calculation, was tested on 1 and 12 threads (see **Sup. Table 1**). The calculation of vectors are clearly faster for the Python code than for the tcl code, this can be in part explained by the use of Numpy[73] to efficiently deal with large arrays in the Python code. So, we believe that there is room for improvement for the tcl code in translating some parts of the calculation in C/C++ or CUDA language. At the inverse the visualization of streamlines is clearly faster using VMD: this is due to an implementation in C in the core of the program which is well optimized in comparison to the Matplotlib/MayaVi code. We can also notice that for our systems the use of multithread calculation with Python only improves the results by a small amount. The VMD visualization takes a little bit longer when high quality pictures (as presented in this article) are rendered using the VMD ray-tracing engine called Tachyon[74].

3.9 Grid size and time scale parameters

We tried different grid sizes for the presented CG models (data not shown). A resolution of 20 Å seems appropriate. This value is coherent with the work of Roark *et al.* defining correlated lipid motions with a length of 25 Å[70]. Furthermore, this resolution value can be linked to the average number of lipid headgroups in each grid cell. This value is 5.7 for the planar membrane. The use of a higher grid resolution of 10 Å resulted in very few lipid headgroups in a given cell, which can defeat the purpose of analysing lipids in unison. Conversely, a lower grid resolution of 40 Å produced reasonable results, albeit less accurate than at 20 Å resolution. So, a minimum 2D system size for use of the streamline visualization with lipids is approximately 200 Å^2 - giving a maximum grid of 100 cells. As we have seen with the planar membrane example system, far greater dimensions are well suited for such a visualization. For the vesicle system, the number of lipid headgroups per cell is 21,5 on average (18 for the outer leaflet and 25 for the inner leaflet). So, we could have diminished the cube size to increase the resolution but we preferred to keep the same grid (20 Å side) to compare the two systems with the same conditions. Furthermore, at this resolution, the MayaVi program has difficulty rendering the scene.

In terms of the time scale over which visualizations are produced, in this paper we have observed that two consecutive frames (0.2 ns) produce an acceptable result. We have not tested our analyses over much longer time scales but we will investigate this in a future study.

3.10 Combining the streamline visualization with other analyses: example with membrane undulations

Correlating diffusional streamlines with visualization of other membrane parameters can be used to highlight subtle dynamic behaviours. In **Fig. 10**, we have colour-mapped the vertical displacement (z coordinate values along bilayer normal) of the membrane in the presence of streamlines. By inspection, the diffusional streamlines seem to evolve in a manner which correlates with the membrane undulations (See **Sup. Movie 5**) and can circulate around vertical undulations of the membrane (see **Fig. 10**). The influence of the surface shape on lipid dynamics has been highlighted by experimental results[75,76]. This could be related to membrane tension and pressure with the less curved parts of the membrane presenting fewer constraints and allowing the lipids to circulate more easily. It will therefore be of interest to examine complex membranes by combining visualization of streamlines with analyses such as the calculation of pressure fields and tension lines[77], area per lipid visualization[78] or lipid phase analysis[79].

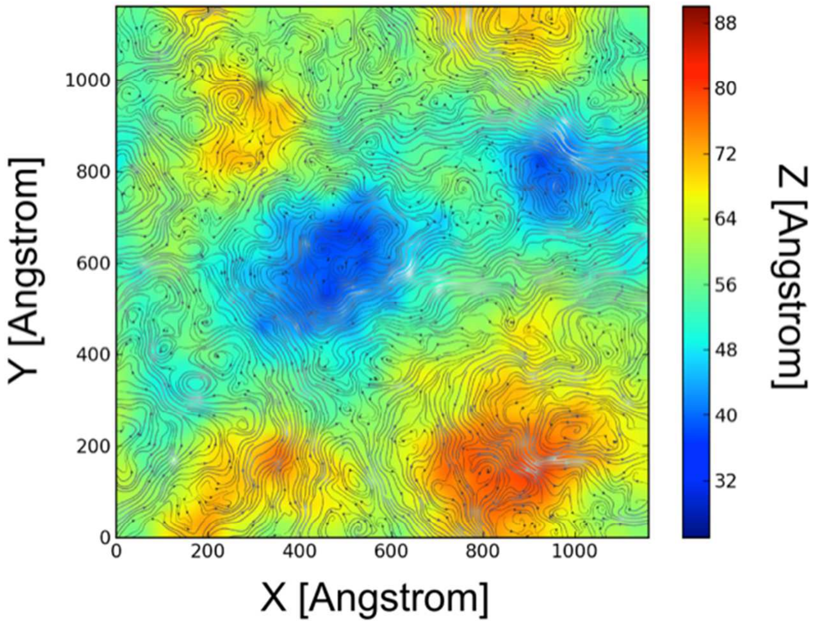


Fig. 10: Streamline visualization coupled with height map for the planar membrane. The colours represent the vertical z values of the membrane. This representation highlights the undulations of the membrane in the z direction. High displacement parts of the streamlines are coloured in white. An associated movie is available in Supplementary Material.

3.11 Future Directions

We have focused this study on the collective behaviour of lipids in bilayer membranes, and are extending this work to analyse the effects of such lipid flows on the diffusion and the clustering of membrane proteins (Goose et al., in preparation). We suggest that these nano-flows can be a more general phenomenon. As the visualisation and analysis method we have described is flexible, it should be possible to apply this approach to other large-scale ensembles of molecules. For example, one might analyse data from simulations of water or solutes (e.g. ions) in the vicinity of transbilayer pores to explore possible nano-flows. In particular, as larger scale membrane simulations become more practicable (and common) it will be important to model such a systems using all-atomistic simulations in order to evaluate the robustness of our approach to different levels of granularity. We also plan to develop a new GPU-accelerated streamline integration and rendering engine for VMD using CUDA, in order to improve the interactive calculation and rendering performances.

4 Conclusion

We have presented a new visualization approach to aid our understanding of the complex dynamic properties of biological membranes. We have illustrated our analysis via two different systems with contrasting lipid compositions, shapes and dynamics. Our methodology has been successfully applied to render lipid motions for these two systems, and allows us to highlight large concerted lipid motions within the plane and between the two leaflets of a membrane. The lipid motions were seen to be linear as well as circular (*i.e.* vortices) in nature and may be linked to membrane deformations perpendicular to the bilayer plane. Finally, we have shown that the leaflet asymmetry in lipid composition can be related to an asymmetry in term of dynamics.

Our set of tools is not intended to solve by itself the complex question of transient lipid flows, but it can be used to focus attention on particular aspects of the dynamic behaviour of membrane prior to more detailed statistical analysis. As larger and more complex lipid bilayer simulations become the norm, we anticipate that these visualization techniques will help us to better define the complexities of lipid movements. This, in turn, will enable improved parameterization of more coarse grained (e.g. DPD) and/or mesoscopic models of biological membranes. If they can accurately capture the key elements of complex membrane dynamics, such higher level models will then allow us to relate molecular properties of complex biological membranes to e.g. super-resolution microscopy imaging of the dynamics of membrane in cells[80].

Scripts are available at the address: http://sbcb.bioch.ox.ac.uk/flows/

Acknowledgements

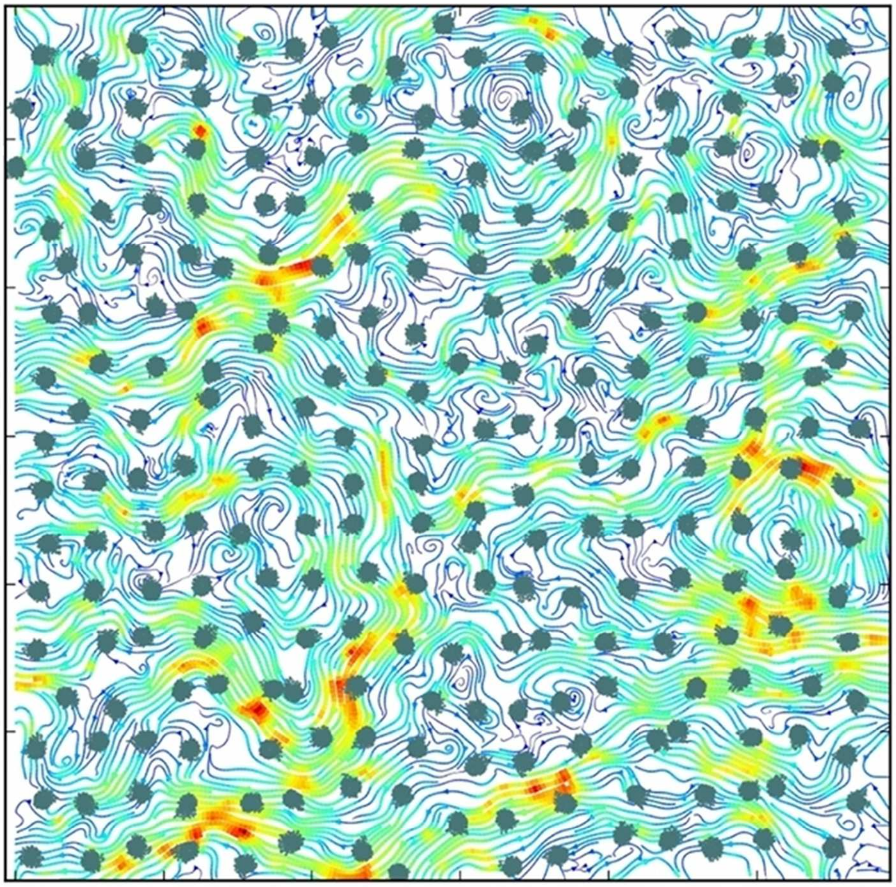
We thank Drs Philip Fowler, Sebastian Busch and Mickael Lelimousin for discussions and Dr Oliver Beckstein for his comments. This work is supported by grants from the Wellcome Trust and the BBSRC. T.R. is supported by a Canadian Institutes of Health Research (CIHR) post-doctoral fellowship and a Fulford Junior Research Fellowship from Somerville College, Oxford. A.C.E.D is supported by an Engineering and Physical Sciences Research Council Systems Biology Doctoral Training Centre studentship. Computational resources were provided in part by the Partnership for Advanced Computing in Europe (PRACE) grant number PP2012061115. VMD development is supported by NIH grant 9P41GM104601.

References

1. S. J. Singer and G. L. Nicolson, Science, 1972, 175, 720–731.
2. D. M. Engelman, Nature, 2005, 438, 578–580.
3. F. Spira, N. S. Mueller, G. Beck, P. von Olshausen, J. Beig, and R. Wedlich-Söldner, Nat. Cell Biol., 2012, 14, 640–648.
4. H. Sprong, P. van der Sluijs, and G. van Meer, Nat. Rev. Mol. Cell Biol., 2001, 2, 504–513.
5. E. J. Denning and O. Beckstein, Chemistry and Physics of Lipids, 2013, 169, 57–71.
6. S. J. Marrink, A. H. de Vries, and D. P. Tieleman, Biochim. Biophys. Acta, 2009, 1788, 149–168.
7. S. M. Loverde, D. A. Pantano, D. A. Christian, A. Mahmud, M. L. Klein, and D. E. Discher, Current Opinion in Solid State and Materials Science, 2011, 15, 277–284.
8. E. Falck, T. Róg, M. Karttunen, and I. Vattulainen, J. Am. Chem. Soc., 2008, 130, 44–45.
9. T. Apajalahti, P. Niemelä, P. N. Govindan, M. S. Miettinen, E. Salonen, S.-J. Marrink, and I. Vattulainen, Faraday discussions, 2010, 144, 411–30– discussion 445–81.
10. S. Busch, C. Smuda, L. C. Pardo, and T. Unruh, J. Am. Chem. Soc., 2010, 132, 3232–3233.
11. P. F. J. Fuchs, Methods Mol. Biol., 2010, 654, 403–421.
12. M. Karplus and G. A. Petsko, Nature, 1990, 347, 631–639.
13. S. J. Marrink and D. P. Tieleman, Chem Soc Rev, 2013, 42, 6801–6822.

14. B. J. Reynwar, G. Illya, V. A. Harmandaris, M. M. Müller, K. Kremer, and M. Deserno, Nature, 2007, 447, 461–464.
15. F. J.-M. de Meyer, M. Venturoli, and B. Smit, Biophys. J., 2008, 95, 1851–1865.
16. E. Lindahl and M. S. P. Sansom, Curr. Opin. Struct. Biol., 2008, 18, 425–431.
17. L. R. Forrest, D. P. Tieleman, and M. S. Sansom, Biophys. J., 1999, 76, 1886–1896.
18. M. K. Sener, J. D. Olsen, C. N. Hunter, and K. Schulten, Proc. Natl. Acad. Sci. U.S.A., 2007, 104, 15723–15728.
19. G. S. Ayton and G. A. Voth, Biophys. J., 2010, 99, 2757–2765.
20. P. L. Freddolino, A. S. Arkhipov, S. B. Larson, A. McPherson, and K. Schulten, Structure, 2006, 14, 437–449.
21. E. Mendez-Villuendas, S. Baoukina, and D. P. Tieleman, Journal of Physics, 2012, 385, 012002.
22. S. I. O'Donoghue, A.-C. Gavin, N. Gehlenborg, D. S. Goodsell, J.-K. Hériché, C. B. Nielsen, C. North, A. J. Olson, J. B. Procter, D. W. Shattuck, T. Walter, and B. Wong, Nat. Methods, 2010, 7, S2–S4.
23. M. Chavent, B. Lévy, M. Krone, K. Bidmon, J.-P. Nominé, T. Ertl, and M. Baaden, Brief. Bioinformatics, 2011, 12, 689–701.
24. M. Krone, J. E. Stone, T. Ertl, and K. Schulten, EuroVis-Short Papers, 2012, 67–71.
25. J. E. Stone, K. L. Vandivort, and K. Schulten, Lecture Notes in Computer Science, 2011, 6939, 1–12.
26. J. E. Stone, K. L. Vandivort, and K. Schulten, UltraVis '13 Proceedings of the 8th International on Ultrascale Visualization, 2013.
27. K. Scharnowski, M. Krone, and F. Sadlo, Computer Graphics and Applications, 2013.
28. H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K.-L. Ma, IEEE Comput. Grap. Appl., 2010, 30, 45–57.
29. J. E. Goose and M. S. P. Sansom, PLoS Comput Biol, 2013, 9, e1003033.
30. M. J. Gerl, J. L. Sampaio, S. Urban, L. Kalvodova, J.-M. Verbavatz, B. Binnington, D. Lindemann, C. A. Lingwood, A. Shevchenko, C. Schroeder, and K. Simons, The Journal of Cell Biology, 2012, 196, 213–221.
31. S. Baoukina, S. J. Marrink, and D. P. Tieleman, Biophys. J., 2012, 102, 1866–1871.
32. K. M. Davies, C. Anselmi, I. Wittig, J. D. Faraldo-Gómez, and W. Kühlbrandt, Proc Natl Acad Sci USA, 2012, 109, 13602–13607.
33. M. Louhivuori, H. J. Risselada, E. van der Giessen, and S. J. Marrink, Proc Natl Acad Sci USA, 2010, 107, 19856–19860.
34. P. J. Bond, C. L. Wee, and M. S. P. Sansom, Biochemistry, 2008, 47, 11321–11331.
35. P. J. Bond and M. S. P. Sansom, J. Am. Chem. Soc., 2006, 128, 2697–2704.
36. L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, J Comput Chem, 2009, 30, 2157–2164.
37. B. Hess, C. Kutzner, D. Van Der Spoel, and E. Lindahl, J. Chem. Theory Comput., 2008, 4, 435–447.
38. S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. de Vries, J Phys Chem B, 2007, 111, 7812–7824.
39. L. Monticelli, S. K. Kandasamy, X. Periole, R. G. Larson, D. P. Tieleman, and S.-J. Marrink, J. Chem. Theory Comput., 2008, 4, 819–834.
40. R. Affentranger and X. Daura, J Comput Chem, 2010, 31, 1889–1903.
41. N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, J Comput Chem, 2011, 32, 2319–2327.
42. D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, J Comput Chem, 2005, 26, 1701–1718.
43. A. Raudino, S. J. Marrink, and M. Pannuzzo, J. Chem. Phys., 2013, 138, 234901.
44. D. Hakobyan and A. Heuer, J Phys Chem B, 2013, 117, 3841–3851.
45. W. Humphrey, A. Dalke, and K. Schulten, J Mol Graph, 1996, 14, 33–8, 27–8.
46. J. D. Hunter, Computing In Science \& Engineering, 2007, 9, 90–95.
47. P. Ramachandran and G. Varoquaux, Computing In Science \& Engineering, 2011, 13, 40–51.
48. I. Casuso, J. Khao, M. Chami, P. Paul-Gilloteaux, M. Husain, J.-P. Duneau, H. Stahlberg, J. N. Sturgis, and S. Scheuring, Nat. Nanotech., 2012, 7, 525–529.
49. T. Róg, M. Pasenkiewicz-Gierula, I. Vattulainen, and M. Karttunen, Biochim. Biophys. Acta, 2009, 1788, 97–121.
50. I. V. Polozov, L. Bezrukov, K. Gawrisch, and J. Zimmerberg, Nat. Chem. Biol., 2008, 4, 248–255.

51. G. S. Ayton and G. A. Voth, Biophys. J., 2004, 87, 3299–3311.
52. S. Busch and T. Unruh, Biochim. Biophys. Acta, 2011, 1808, 199–208.
53. M. Javanainen, L. Monticelli, J. Bernardino de la Serna, and I. Vattulainen, Langmuir, 2010, 26, 15436–15444.
54. Y. Andoh, S. Okazaki, and R. Ueoka, Biochim. Biophys. Acta, 2013, 1828, 1259–1270.
55. M. Javanainen, H. Hammaren, L. Monticelli, J.-H. Jeon, M. S. Miettinen, H. Martinez-Seara, R. Metzler, and I. Vattulainen, Faraday discussions, 2013, 161, 397–417.
56. I. Vattulainen and T. Róg, Cold Spring Harb Perspect Biol, 2011, 3.
57. J. K. Sigurdsson, Y. Wang, E. Brandt, and P. J. Atzberger, Phys. Rev. E, 023301.
58. A. R. Honerkamp-Smith, F. G. Woodhouse, and V. Kantsler, Physical review Letters, 2013, 038103.
59. J. J. Tyson, K. Chen, and B. Novak, Nat. Rev. Mol. Cell Biol., 2001, 2, 908–916.
60. J. K. Weber and V. S. Pande, J. Chem. Phys., 2013, 138, 085103.
61. J. K. Weber and V. S. Pande, J. Chem. Phys., 2013, 138, 165101.
62. T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, 2010, 29, 1807–1829.
63. B. Jobard and W. Lefer, Visualization in Scientific Computing'97, 1997, 43–55.
64. N. A. Baker, Proceedings of the National Academy of Sciences, 2001, 98, 10037–10041.
65. S. Vassiliev, P. Comte, A. Mahboob, and D. Bruce, Biochemistry, 2010, 49, 1873–1881.
66. P. S. Niemelä, M. S. Miettinen, L. Monticelli, H. Hammaren, P. Bjelkmar, T. Murtola, E. Lindahl, and I. Vattulainen, J. Am. Chem. Soc., 2010, 132, 7574–7575.
67. H. Yu, C. Wang, C.-K. Shene, and J. H. Chen, Visualization and Computer Graphics, IEEE Transactions on, 2012, 18, 1353–1367.
68. M. Schlemmer, I. Hotz, B. Hamann, F. Morr, and H. Hagen, VGTC Symposium on Visualization, 2007, 227–234.
69. H. Morhenn, S. Busch, H. Meyer, D. Richter, W. Petry, and T. Unruh, Physical review Letters, 2013, 111, 173003.
70. M. Roark and S. E. Feller, J Phys Chem B, 2009, 113, 13229–13234.
71. S. J. Marrink, O. Berger, P. Tieleman, and F. Jähnig, Biophys. J., 1998, 74, 931–943.
72. V. Gapsys, B. L. De Groot, and R. Briones, J. Comput. Aided Mol. Des., 2013, 10, 845–858.
73. T. E. Oliphant, Computing In Science \& Engineering, 2007, 9, 10–20.
74. J. E. Stone, University of Missouri-Rolla.
75. A. Roux, D. Cuvelier, P. Nassoy, J. Prost, P. Bassereau, and B. Goud, EMBO J., 2005, 24, 1537–1545.
76. R. Parthasarathy, C.-H. Yu, and J. T. Groves, Langmuir, 2006, 22, 5095–5099.
77. P. M. Kasson, B. Hess, and E. Lindahl, Chemistry and Physics of Lipids, 2013, 169, 106–112.
78. G. Lukat, J. Krüger, and B. Sommer, J Chem Inf Model, 2013, 53, 2908–2925.
79. E. Jefferys, M. S. P. Sansom, and P. W. fowler, Faraday discussions, 2014.
80. C. Eggeling, C. Ringemann, R. Medda, G. Schwarzmann, K. Sandhoff, S. Polyakova, V. N. Belov, B. Hein, C. von Middendorff, A. Schönle, and S. W. Hell, Nature, 2009, 457, 1159–1162.



Lipid motions patterns highlighted by the streamline visualization.
95x92mm (199 x 199 DPI)