Sequential Voxel-Based Leaflet Segmentation of Complex Lipid Morphologies

Bart Bruininks, Albert Thie, Paulo Souza, Tsjerk Wassenaar, Shirin Faraji, Siewert Marrink

Sequential voxel-based leaflet segmentation of complex lipid morphologies

Bart M. H. Bruininks,[*,†] Albert S. Thie,[‡] Paulo C.T. Souza,[¶,†] Tsjerk A. Wassenaar,[†] Shirin Faraji,[‡] and Siewert J. Marrink[†]

†*Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, Groningen*

‡*Zernike Institute for Advanced Materials, University of Groningen, Groningen*

¶*Molecular Microbiology and Structural Biochemistry (MMSB, UMR 5086), CNRS, University of Lyon, Lyon, France*

E-mail: b.m.h.bruininks@rug.nl

Abstract

As molecular dynamics simulations increase in complexity, new analysis tools are necessary to facilitate interpreting the results. Lipids, for instance, are known to form many complicated morphologies due to their amphipathic nature, becoming more intricate as particle count goes up. A few lipids might form a micelle, where aggregation of tens of thousands could lead to vesicle formation. Millions of lipids make up a cell and its organelle membranes, and are involved in processes such as neurotransmission and transfection. To study such phenomena, it is useful to have analysis tools which understand what is meant by emerging entities such as micelles and vesicles. Studying such systems at the particle level only, becomes extremely tedious, counter-intuitive and computationally expensive. To address this issue we developed a method to track all the individual lipid leaflets, allowing for easy and quick detection of topological

changes at the meso scale. By using a voxel-based approach and focussing on locality, we forgo costly geometrical operations without losing important details and chronologically identify the lipid segments using the Jaccard index. Thus we achieve a consistent sequential segmentation on a wide variety of (lipid) systems, including monolayers, bilayers, vesicles, inverted hexagonal phases, up to the membranes of a full mitochondrion. It also discriminates between adhesion and fusion of leaflets. We show that our method produces consistent results without the need for pre fitting parameters, and segmentation of millions of particles can be achieved on a desktop machine.

Introduction

The increase in complexity of molecular dynamics (MD) simulations poses challenges for current analysis software. This is well illustrated in the field of biological applications. Such simulations changed from a single protein in vacuum [1] in the late 1970s to complex membrane based systems containing millions of molecules such as (phospho)lipids, cholesterol, proteins, DNA, drugs, water, etc.[2–7] The number of molecular aggregates in simulations has also increased, and nowadays it is not rare to see a mixture of micelles, bilayers and vesicles in a single simulation.[4,8–11] This increase in complexity of both composition and aggregation state requires a higher level of abstraction in analysis than typically offered. During analysis we would like to work with concepts such as bilayers, vesicles, leaflets, protein aggregates, phases, etc. on top of simple identifiers such as molecule names and indices, often used in current tools. This would dramatically reduce the complexity of coding the analysis software and would allow for a more user friendly interface with data. Current MD analysis software, such as MDAnalysis,[12] Pytim,[13] FATSLiM[14] and VMD,[15] come with built-in functions or have packages to detect e.g. bilayer leaflets and aggregates, and offer some of the desired level of abstraction. However, these tools were developed to deal with relatively simple compositions and topologies, and may fail as systems contain more than one or two segments (SI Fig. 8). To avoid this problem we developed a robust method for consistent sequential segmentation

with a voxel oriented analysis paradigm. No assumptions are required regarding the number of segments or molecules, and it natively supports all-atom or coarse-grained (CG) simulation data. Furthermore, voxel-based methods offer very good scaling with the increase of particles and are excellent for locality queries.[16–19]

To guide the reader through this work we will first present pseudo code to illustrate the general implementation of our leaflet segmentation algorithm, which is implemented in a tool coined MDVoxelSegmentation. We then show a wide range of examples, and we end with a discussion of the limitations and further prospects of the tool. The Supporting Information (SI) contains the video results of the algorithm.

Design and implementation

In the following sections, we describe the complete algorithm in two sections of pseudo code. We start with considering a single frame and perform our multi-step connective components analysis for lipid-based systems. Then we describe how we add the extra layer of chronological consistency on top.

Pseudo code for spatial segmentation of leaflets

The aim is to segment a lipid density into leaflets (i.e. a single membrane leaflet is a segment). To successfully segment lipid densities into leaflets we require three things, a reference, a trajectory and a selection file. The reference file should contain information regarding atoms and their naming (e.g. GRO, PDB). The trajectory file should contain coordinates for each point in the reference file in every frame (e.g. GRO, XTC, TRR). Finally, the selection file should contain the selections (lipid heads, lipid tails, lipid linkers and exclusion beads). By default most CG Martini lipids are supported,[20] meaning no selection has to be specified for the leaflet segmentation of those systems. However, if you are working on a custom lipid, it can easily be added to the default selections file (an example for CHARMM lipids is given in

the methods for the leaflet segmentation). The completed segmentation will assign each lipid (residue) a segment identity. A schematic representation of leaflet segmentation is shown in Fig. 1.

Leaflet segmentation starts with reading the input files (Fig. 1 A) and mapping all the lipid tails, lipid heads, and exclusion beads to boolean voxel matrices (0.5 nm binning for Martini lipid systems; Fig. 1 B). This results in the following three 3D boolean matrices: tails, heads and exclusions. The exclusions are always grown once, meaning all neighboring voxels (26) of a True voxel become True. The exclusions are used as local stops during segmentation, acting as boundaries for the segmentation (i.e. exclusions can be used to confine the segmentation). We found that using exclusions can be beneficial when working with proteins, since the protein boundaries are prone to be edge cases for lipid segmentation and are better handled later.

The tails matrix is segmented using the adapted connected components algorithm (Fig. 1 C). Before segmentation starts, the heads matrix is subtracted from the tails matrix. The exclusions act as boundaries during segmentation. The result of the tails segmentation is depicted in Fig. 1 D.

Roughly the same procedure is followed for segmenting the heads as was done for the tails. However, instead of using the complete heads matrix, it is first deconstructed by the tails segments. Each tails segment is individually processed (Fig. 1 E, G). This is achieved by using only those headgroups which have a lipid residue index which is also in the active tails segment. Before heads segmentation starts, the tails matrix is subtracted and the exclusions act as boundaries during segmentation. Segmenting the heads in this manner results in the leaflet segmentation per tails segment (Fig. 1 F, H).

Although the result at this point is good (Fig. 1 I, J), there might be some lipids which still have no segment assigned. Force-segmentation can solve this problem as it can be used to map lipids to a segment based on a distance criterion. For every lipid which does not have a segment assigned, all neighbouring lipids are checked within a certain cutoff distance

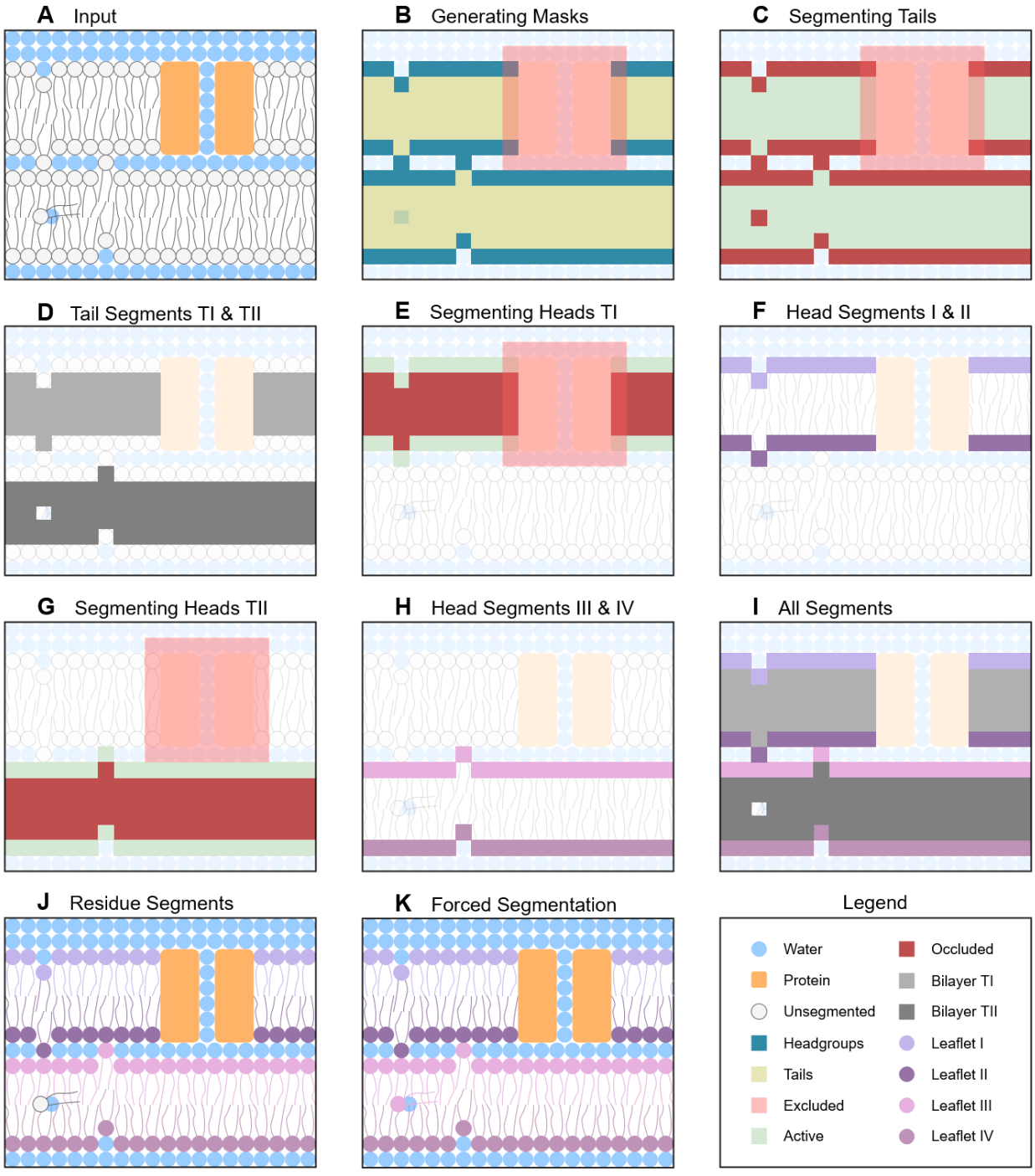

Figure 1: Spatial leaflet segmentation. A 2D schematic representation of the leaflet segmentation of two stacked bilayers in a periodic box. The upper bilayer contains a protein channel and the lower bilayer contains a diving lipid. The protein is used as exclusion selection.

(default is 2 nm). If the dominant segment in that search is not 0 (unsegmented), the lipid is appointed to that dominant segment. Our implementation performs this operation in an iterative manner by trying to classify with small cutoff distances first (default is 1 nm). The cutoff is slightly increased (default is 0.1 nm) if no extra lipids were segmented. If an iteration does result in new lipids being segmented the cutoff is set back to the minimum value. The iteration halts if the force-segmentation returns a stable output (i.e. no further changes occur), if the cutoff becomes larger than the maximum value, or if all selected lipids have been segmented. The final result after force-segmentation is depicted in Fig. (1 K). Force-segmentation can be turned off.

Splitting the segmentation into two steps, using force-segmentation as a post processing step, means that simple cases are handled in a relatively low-cost manner where hard cases might use a different approach altogether (in this case our force-segmentation protocol). This mixed use of a low-cost algorithm with more precise and expensive algorithms reduces the total complexity of the calculation, whilst at the same time hardly affecting the segmentation quality.

The result after spatial leaflet segmentation of a trajectory, is a new trajectory which contains a segmentation entry for each index in each frame. These indexes correspond to the atom indices in the original structure file. Atoms which have no segmentation are assigned to 0. However, due to the arbitrary picking of our initial voxels for segmentation, the segmentation labels miss sensible consistency over time for all other labels than 0. To solve this lack of chronological consistency, we need to perform temporal segmentation.

Pseudo code for temporal segmentation

All that is required as input for the temporal segmentation is the spatial segmentation output. To perform temporal segmentation, we need to find the identity of segments for each frame sequentially. When comparing a new frame with a current frame, the algorithm first considers the identities of segments of the current frame. The identified segments of the current frame

are then matched to the segments in the new frame (Fig. 2 A). The atoms within a segment are used as the basis for the identity of the segment. Segments are treated as sets, for which the unique elements are the atom indices. The identity of a segment in the new frame is the segment in the current frame with which it shares the most atoms. In this way segments can be matched across frames, even if due to shrinking or growth their atom contents are not identical. Consider frame N which contains two segments $N1 = \{a1, a2, a3, a4\}$ and $N2 = \{a5, a6, a7, a8\}$ and frame M which contains the segments $M1 = \{a1, a2, a3, a8\}$ and $M2 = \{a5, a6, a7, a4, a9\}$. Each segment contains unique elements, which have a lowercase unique code. When we compare frame N with frame M, we see that the elements $a8$ and $a4$ have swapped places and $a9$ is introduced to M2. However we still want to assign N1 to M1 and N2 to M2, to allow for the growing and shrinking of segments. We therefore need a way to identify the amount of shared elements between two segments.

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

To identify how many atoms are shared between two segments across frames the Jaccard index is used (eq. 1). This index is defined as the intersection of two sets divided by the union of two sets. This provides a ratio between 0 and 1, where 1 is complete similarity and 0 means no shared atoms. The parameter Jaccard threshold defines the minimum score on the Jaccard index that is required for two sets to have the same identity (SI Fig. 10). For example a Jaccard threshold of 0.618 means that only if two segments have a score higher than 0.618 together, they have the same identity. This allows for consistency while also identifying large shifts between frames in the simulation. For example a leaflet in a bilayer system which has lost lipids due to flip-flopping can still be indentified in the next frame. Because of the possibility of these large shifts the algorithm does not assume that the amount of segments between frames remains the same. It therefore allows for the disappearance of identified segments and the appearance of new identities. Identified segments can disappear when no match in the new frame can be made that exceeds the Jaccard threshold. An

identified segment disappearing is considered to have merged with another segment. A new identity can be created when in the new frame a segment exists to which no old identified segment has matched. This new segment then receives a new identity. A newly created identified segment is considered to have split from a previously existing identified segment.

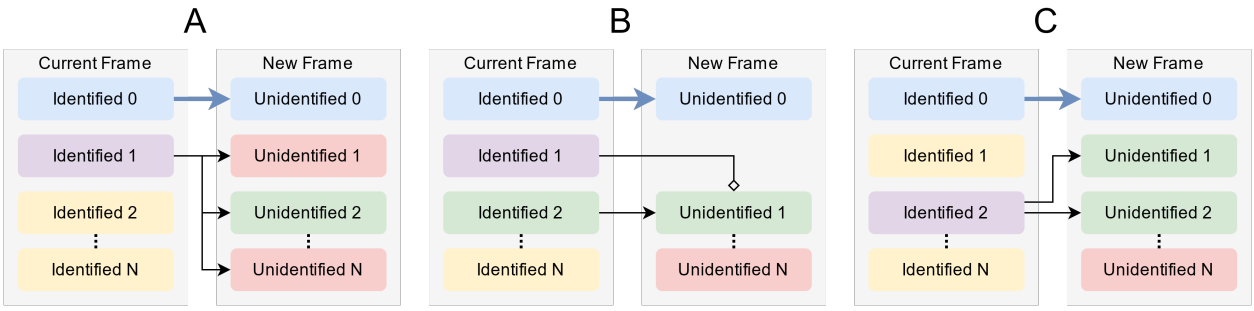

Figure 2: Finding the identity of the segments in the new frame. (A) The ordered 1 segment has a best match with unordered 2. (B) The ordered 1 and 2 are merged into one segment, with ordered 2 providing the identity for unordered 1. (C) The ordered 2 is split into two new labels, unordered 1 and 2. Ordered 0 is always matched as the new segment zero, as this is the segment which contains all non-segmented particles. Capital N indicates the total number of segments in each frame.

In the case of a merge (Fig. 2 B), two or more identified segments in the current frame are merged into an unidentified segment in the new frame. Two distinct cases are possible in the new frame. Either one of the identities of the merging segments is inherited by the new segment or a new identity is created. If one of the identities is maintained, the Jaccard score between one of the identified segments and the unidentified segment has exceeded the Jaccard threshold. This indicates a large segment is merging with one or more smaller segments. For example, this could be the result of a transient fusion between two leaflets. If a new identity is created, this means none of the previous segments contained enough atoms to attain a Jaccard score higher than the Jaccard threshold when compared to the unidentified segment. In this case all previous identified segments are merged into a new segment with a new identity.

To allow for the reappearance of identities, the disappeared identities are stored in a database. This database contains the identity and the atoms present in the identified segment

at the current frame. It also logs into which segment the disappeared segment has merged. The segment into which the disappeared segment has merged is the segment for which it had the highest Jaccard score. This score will always be below the Jaccard threshold, as otherwise the identity would be inherited. This information can be used to restore identities when splitting events take place.

In a split, two or more unidentified segments in the new frame have the highest Jaccard score with the same identified segment in the current frame (Fig. 2 C). In such a split there are also two distinct cases. In one case a small part splits off from an existing identified segment. However one of the unidentified segments still has a high enough Jaccard score to match with the identified segment. This means that one of the unidentified segments inherits the identity of the identified segment. This could indicate a leaflet loosing a small group of lipids which form their own segment. The other unidentified segments will then either gain a new identity or take the identity of an old segment from the database. The new segment takes the identity of the segment with the highest Jaccard score in the database, but only if the score exceeds the Jaccard threshold. If the threshold is not exceeded, the segment is assigned a new identity.

In the other case the identified segment could not match with one of the unidentified segments. This will lead to the disappearance of the identified segment. This could be caused by a leaflet splitting into two roughly equal parts. For example, in the case of a bilayer with a transient pore, closing of the pore would result in a split of the bilayer in its previous two leaflet segments. The disappearing segment will be stored in the database, while the new segments will each be compared to the database. Each segment which has a match with the database will continue with the identity of this match, while the unmatched segments continue with new unique identities.

The result after temporal segmentation is a consistent sequential segmentation over the trajectory, both spatially and over time. Identities between time frames are consistent, based on the atoms within each segment. Merging segments are detected and identified in a size

dependent way. Splitting segments are detected as well. The database of previous segments allows identities to persist over a longer time frame.

Results

To validate the leaflet segmentation algorithm, we prepared a wide range of lipid systems using the Martini CG force field.[21] The same default input was used for all segmentation unless specifically stated otherwise. We start with simple lipid bilayer systems in varying topologies (Fig. 3). Then we add proteins and a high variety of compositional complexity (Fig. 4). This is followed by a complex phase transition and fusion of DNA-lipid complexes (lipoplexes) with endosome membrane models to illustrate segmentation quality in the presence of tight, curved and complex dynamic geometry (Fig. 5). We end with some more general real world examples, including an example at atomistic resolution (Fig. 6).

Pure bilayer systems

Leaflet assignment for a simple bilayer containing 270 DOPC and 66 cholesterol (4:1) without force-segmentation resulted in good segmentation, but some lipids, in particular cholesterol, are unlabeled (Fig. 3 A, C orange). By turning on force-segmentation all lipids including the cholesterol could be assigned successfully to a leaflet (Fig. 3 B, C green).

A common lipid property to investigate is the flip-flop rate. We compared the assigned flip-flopping of our leaflet segmentation to a common procedure. This common analysis would assign a lipid flip-flop based on the z-coordinate of a lipid headgroup with respect to the average z-position of the bilayer. To prevent high frequency noise due to headgroups at or close to the membrane center, a deadzone is used. Lipids within the deadzone region around the center are not assigned to a leaflet. The downside of the z-height approach is that it can only be used for small patches of a relatively flat bilayer with its normal pointing along the z dimension. Our method takes a rather different approach. We check for changes

in leaflet membership for each lipid. To reduce noise, a lipid which is not segmented is assigned to its previous valid leaflet. The advantage of our segmentation based method over the conventional approach is that it still works if the membranes/leaflets are curved, or if they are not a well defined bilayer at all. In our comparison of cholesterol flip-flopping, our algorithm performs well which is indicated by the close relation between the segmentation labels and the discretization of the lipid z-height (Fig. 3 C). Segmentation without force-segmentation is good enough for lipid flip-flop calculations. The segmentation based flip-flop analysis compares well with the z-height approach if a deadzone of ± 0.6 nm is used in the latter (Fig. 3 D). Although the flip-flop patterns for both methods are highly similar, they are not the same, therefore, direct quantitative comparison between both methods should be performed with care. However, if a z-height deadzone of ± 0.3 nm is used, nearly all flip-flops indicated by the segmentation analysis are in the set of the z-height approach (SI Fig. 9). In other words it appears that the flip-flops indicated by the segmentation are a near perfect subset of the z-height approach with a relatively small deadzone.

Since stacked bilayers are notoriously hard to label correctly in other methods, we designed a test system containing two closely stacked and intercalating bilayers of POPC (on average two hydration layers of CG water beads, Fig. 3 E). The close proximity and or intercalation of the bilayers did not hinder correct leaflet segmentation at all over a trajectory spanning 10 µs (Fig. 3 F).

The final pure bilayer system tested is a bilayer in which we artificially opened and closed a pore using a biasing potential. We used the leaflet segmentation to detect the instances of pore opening and closing (Fig. 3 G and H, SI video 1).

Consistent leaflet segmentation was successfully obtained using the default settings for all pure bilayer systems even in the presence of cholesterol or intercalating headgroups. The leaflet segmentation can also be used for flip-flop analysis and the detection of a toroidal pore with high accuracy.

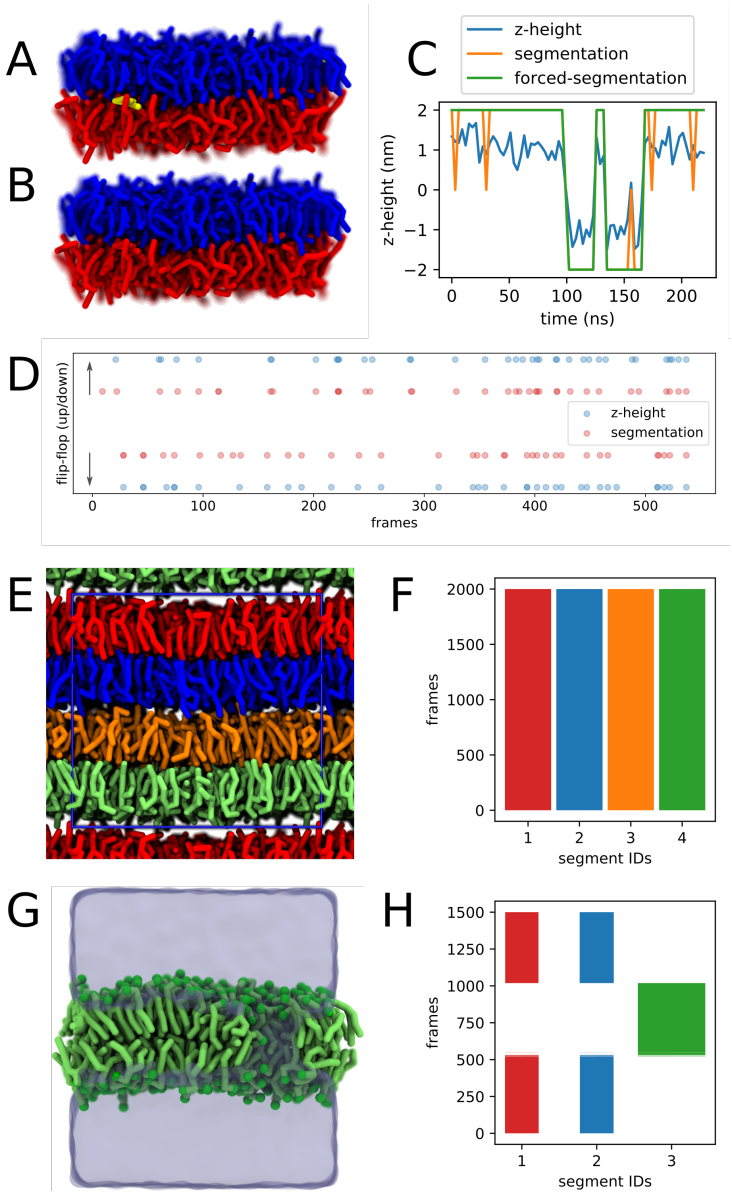

Figure 3: Leaflet segmentation and analysis for pure bilayer systems. (A, B) Snapshots of a DOPC/CHOL(4:1) bilayer with and without force-segmentation. Some lipids were unassigned without force-segmentation due to diving/occlusion (A, yellow). Using force-segmentation non-segmented lipids are assigned to a segment as a post-processing step (B). (C) The flip-flop traces of a single cholesterol in the DOPC/CHOL bilayer, comparing the z-height and segmentation flip-flop approaches. Whenever the orange line is not visible it lies perfectly behind the green line. (D) A comparison of the up- and downward cholesterol flip-flop events between the z-height and segmentation based flip-flop analyses. A more intense single dot represents more than one cholesterol flip-flopping at that given instant. (E) A snapshot of closely stacked and intercalating bilayers of DLPC. (F) Leaflet segmentation of the intercalating bilayers over 10 µs. The width of a segment represents its relative size (G) A snapshot of a DLPC bilayer with a toroidal pore (frame 750 of H). (H) Leaflet segmentation of pore formation. The pore forming biasing potential was turned on between frame 500 and 1000. The width of a segment represents its relative size.

Complex bilayers with proteins

After confirming that our leaflet segmentation performs well on simple lipid-only bilayer systems, we went one step further and considered systems of higher compositional complexity. As for the lipid only segmentation, segmentation of a simple bilayer containing POPC lipids and a single protein is perfect using the default settings (Fig. 4 A, SI video 2). Next, we tested a crowded plasma membrane model. This model contains a high variety of lipids ($\sim$ 60) and a wide range of proteins in a planar membrane.[6] Segmentation resulted in a nearly perfect assignment (Fig. 4 B). However there are 12 lipids which remain unlabeled (Fig. 4 B zoom box), caused by an exclusion region. Setting the force-segmentation cutoff radius slightly higher resolved the issue, but the current result is shown to demonstrate the quality of the default settings. Missing 12 lipids out of tens of thousands of lipids would probably not affect the analysis to any serious degree. However, if perfection is required some tweaking might be needed. For example the force-segmentation range or the binning distance could be altered as well as the minimum cluster size.

Based on these results we conclude that MDVoxelSegmentation works well on systems with a complex composition even in the presence of protein(s).

Curved and tight geometry

Up to now all our test systems were relatively flat, therefore as our final test set, we introduce heavy curvature as well as closely packed segmentation. In our previous work we showed the phase transition of a stacked bilayer system to an inverted hexagonal phase (H_{II}),[4,22] which was used as a system which should push the time consistency part of our algorithm, as well as the agonistic behaviour with respect to the amount of segments. A snapshot of the first intermediate and last frame of the trajectory is shown in Fig. 5 A. The corresponding graph shows that the segmentation is extremely stable over time. A link to the video of the full phase transition is given in the SI as video 3. During transition the following happens: First the leaflets which share a common water reservoir (Fig. 5 A I) fuse together by multiple

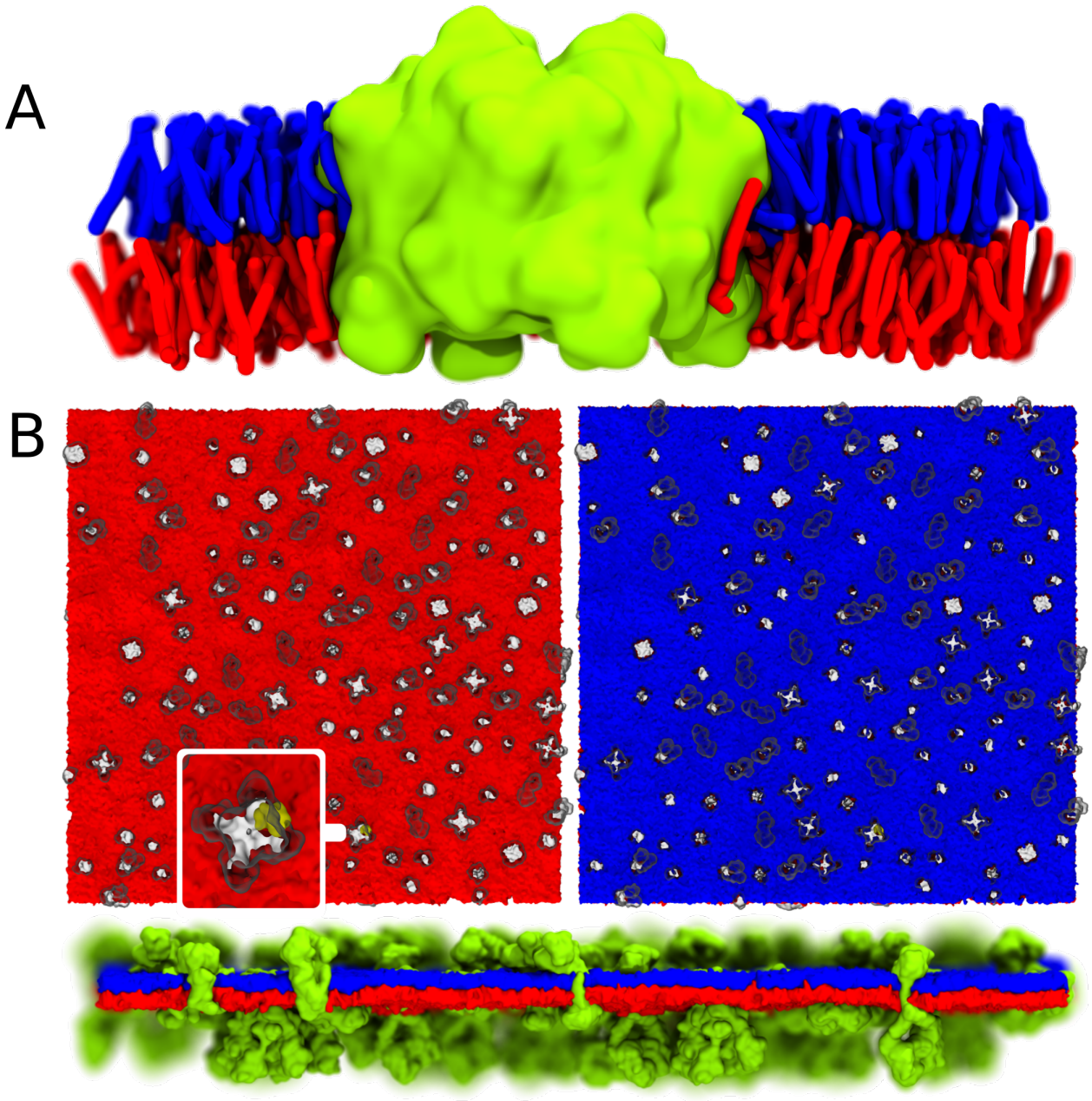

Figure 4: Leaflet segmentation in the presence of protein. (A) Adding protein (lime) to a bilayer does not interfere with the leaflet assignment of a CG POPC bilayer (red/blue). A video of the bilayer protein system can be found in the SI. (B) The bottom and top leaflet of the plasma membrane. Even extremely complex bilayers such as the plasma membrane including many different proteins (transparent/lime) and lipids (red/blue) do not hinder leaflet assignment. However, there is a small cluster of non-assigned lipids (yellow) around one of the proteins (zoom box).

stalk formation (Fig. 5 A II). The leaflets are then split into smaller segments, more closely resembling the classic view of the H_{II} phase. However, due to the high concentration of water in this system not all aquatic channels are singular and some remain connected through water pores perpendicular to the view normal (Fig. 5 A III). The final result is a stable H_{II} phase of channels with a varying degree of interconnection. Note that the visualization of the segments (which is part of the output, as can be seen in the Method section) immediately makes it clear which channels are connected and which not.

In our previous work we showed how transfection with lipoplexes can be simulated using the Martini force field.[4] Such lipoplexes are complexes of lipids and dsDNA. The dsDNA resides inside the inverted hexagonal phase of the lipoplex lipids. Upon fusion with a membrane the dsDNA is released into the cytosol. The tight packing of the lipids with the dsDNA in combination with the change in topology during transfection, makes this another challenging system to analyse. A render of the leaflet segmentation is shown in Fig. 5 B. A link to the video of the complete transfection process is available in the SI video 4. Although the segmentation is pretty good, it does contain some noise, illustrated by minor flickering in the video. This noise could be removed with a noise filter, but we left it in for fair comparison and to create a realistic reference for the quality of the output under comparable conditions. As for pores in the bilayer, we can use the amount of leaflet segments identified in the simulation as a reliable metric for fusion events (Fig. 5 B, graph). We did find that decreasing the resolution of the leaflet segmentation for complex geometry resulted in a steep decrease in quality of segmentation. For more simple geometry 1 nm voxels without hyper-resolution resulted in acceptable segmentation. However, for these systems 0.5 nm voxels with hyper-resolution is required.

The presence of (high) curvature does not hinder correct leaflet segmentation, even if the lipid arrangement does not follow a strict bilayer definition or if the amount of segments is highly dynamic.

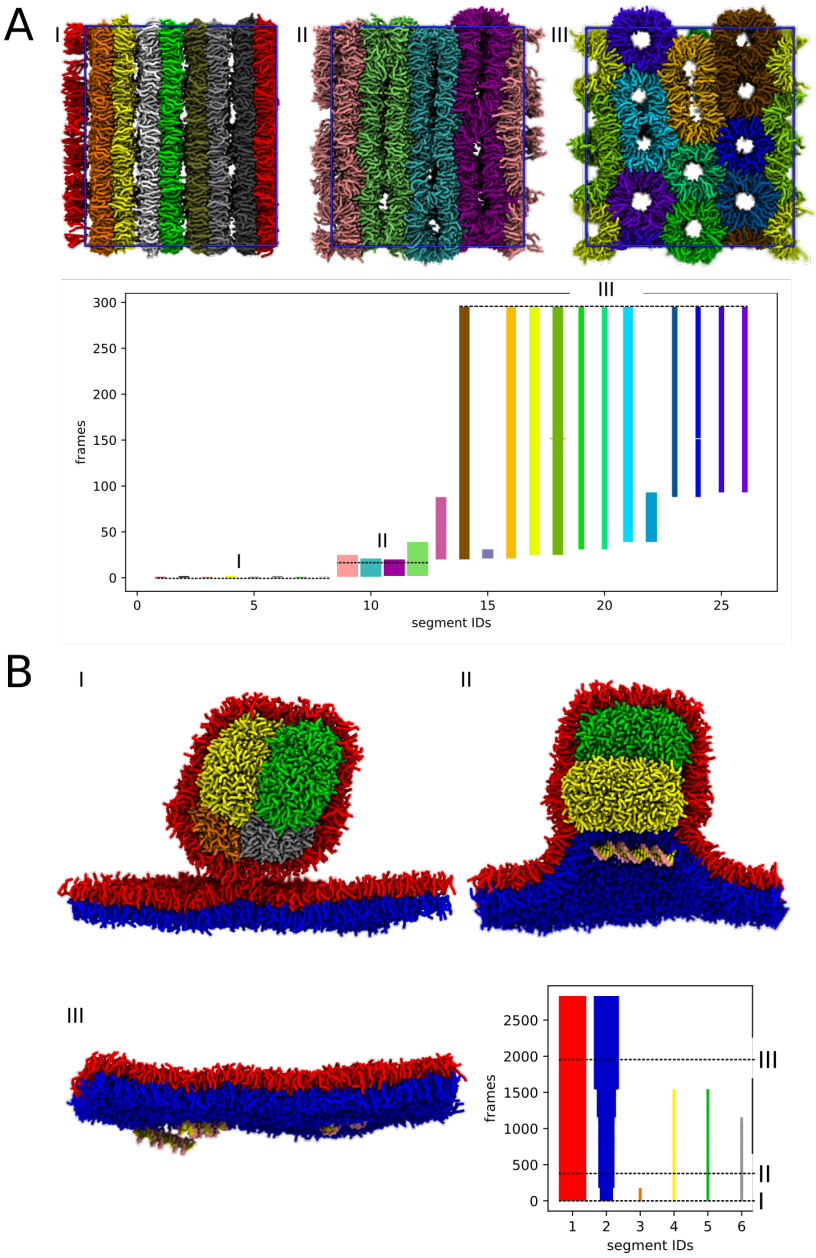

Figure 5: Leaflet segmentation and analysis for both tight and curved complex geometry. (A) The phase transition from a multilamellar bilayer system to an H_{II} phase. Snapshots of the first, intermediate and last segmented frame of a 1 µs trajectory. The accessory video is available in the SI at supporting video 3. The corresponding graph shows all segments formed in chronological order with the snapshots indicated with the dotted lines. The width of a segment indicates its relative size. (B) Snapshots of a lipoplex transfecting dsDNA over a bilayer (first, intermediate and last segmented frame). The inner core (orange, yellow, green, gray) is an intact H_{II} phase containing the dsDNA (not visible), whereas the upper bilayer leaflet (red) has already fused to the outer leaflet of the lipoplex by means of a fusion stalk (I). During transfection the inner core segments fuse with the bottom leaflet of the bilayer (II), resulting in a single flat bilayer with all the dsDNA on the other side (III). The transfection can be tracked in detail using the segmentation graph. The width of a segment indicates its relative size.

Further tests with complex and real world systems

The leaflet segmentation was tested on three additional complex systems to obtain familiarity with the quality. The first system contains a small fatty acid vesicle (Fig. 6, SI video 5). Around the vesicle fatty acids are suspended in a water buffer. The goal was to measure the imbalance of molecules in the inner and outer leaflet of the vesicle as it absorbs the surrounding fatty acids in solution. As the outer leaflet of the vesicle merges with more and more small micelles in solution the imbalance increases, potentially resulting in division. Some flip-flopping occurs to release the tension due to the imbalance between the inner and outer leaflet. However, there are two moments at which the inner and outer leaflet of the vesicle temporarily fuse by means of a toroidal pore. These pores quickly cause a significant amount of acyl chains to migrate from the outer to the inner leaflet, indicated by a jump in the inner leaflet's size (Fig. 6 A). Passive flip-flop occurs all the time and is indicated by the more continuous increase in size of the inner leaflet. This example furthermore illustrates the use of our segmentation tool to track flip-flop and pore formation events with high resolution.

The second test was to assign the leaflets in the showcase mitochondrion model presented in the work by Pezeshkian et al.[10] For the mitochondrion, we expected to observe four separated leaflets, namely the separate leaflets of both the inner and outer membrane. However, after leaflet segmentation of the complete mitochondrion we only observed three segments. To further investigate this issue we cut the mitochondria in twenty slices. Each piece was segmented individually (Fig. 6 B). This revealed that there are issues in two of the twenty slices. When these two problematic slices are excluded the expected four leaflets were found by segmentation analysis. This indicates unexpected overlap is occurring in those two slices, likely as a result of stalk formation between closely apposed membrane segments. Further subdivision could be utilized to pinpoint the exact location, although doing so by hand becomes rather cumbersome. Nevertheless this example shows that large structures can be successfully handled by our algorithm and the results can be used for quality control of systems too large for visual inspection by traditional means.

Finally, in our previous publication on lipoplexes we described the existence of connective channels in an overly hydrated inverted hexagonal lipid phase, shown in the example above on formation of an H_{II} phase(Fig. 5 A).[4] These channels probably have huge effects on the fusion kinetics of such phases. Related work was performed at atomistic scale by Ramezanpour et al., in which they did not describe any of such channels.[8] After contacting the group and presenting this paradox to them, we were kindly allowed access to their data. We used MDVoxelSegmentation to find any possible unexpected and undescribed connective channels. The channels were confirmed to also exist in the atomistic simulations, mostly forming at the mid to end of the trajectory of their lipid system with the highest hydration (Fig. 6 C). The channel was stable, but difficult to spot using the conventional tools available. This due to the fact that it spanned over the PBC, a common problem in periodic systems. MDVoxelSegmentation was able to classify the channel, as it supports all PBC used for molecular simulations. Checking the marked frames by eye confirmed that the results of MDVoxelSegmentation were correct, the authors agreed after reevaluating their data. The formation of these pores was not observed at lower temperatures, suggesting that there is a kinetic factor and that their simulations might benefit from extension.

Together, these examples show that MDVoxelSegmentation is a fast and useful tool for spotting complex segmentation properties, otherwise easily missed or extremely cumbersome to analyse.

Performance

To get some insight in the performance of the current MDVoxelSegmentation tool we timed the segmentation for some of the test systems using a varying amount of threads. The amount of threads were varied to test the parallel processing of MDVoxelSegmentation. In all these tests we turned off force-segmentation, for it can have a large impact on performance based on the exact system (especially at higher maximum recursion depth). The results are summarized in Fig. 7. From the single thread values we can deduce that our algorithm

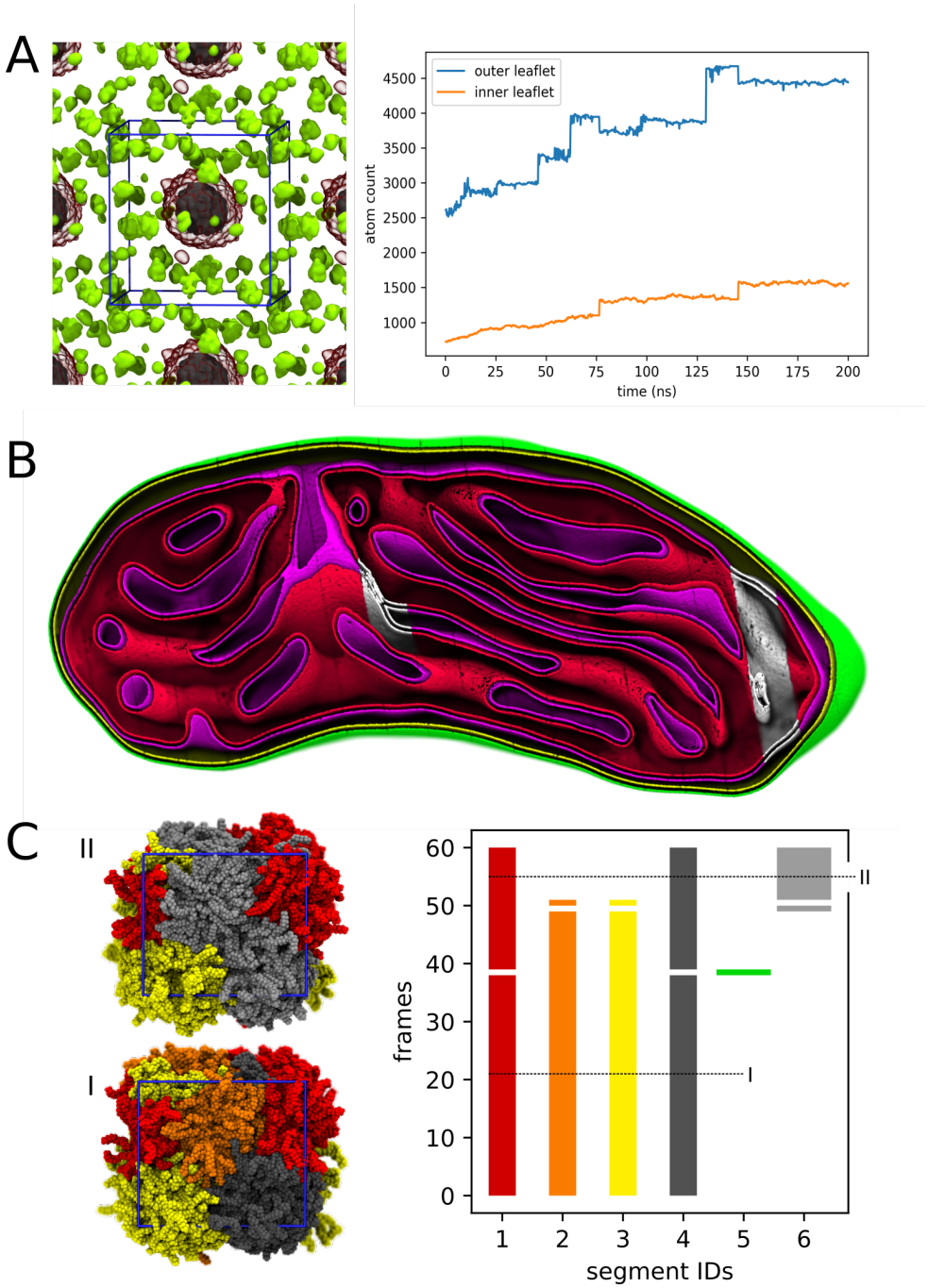

Figure 6: Additional systems. (A) A snapshot of a fatty acid vesicle (red/grey), surrounded by individual fatty acids in the medium (green). The amount of fatty acids in the inner and outer leaflet of the vesicle over time (graph). A video of the fatty acid dinner is available in SI supporting video 5. (B) A snapshot of the segmented mitochondrion (˜83 million beads). Four distinct leaflets can be traced (green, yellow, purple, red) if the intersection zones are not included in segmentation (white). If the white zones are included in the segmentation, the pink and red leaflets are fused together. (C) Snapshots of the atomistic POPC system in the H_{II} phase as previously published.[8] Segmentation analysis reveals previously unnoticed connections between the initial four channels at the end of the simulation. The width of a segment in the graph indicates its relative size.

indeed scales pretty much linearly with some variance due to packing and complexity (table 1). Systems containing thousands to millions of particles of interest will roughly take between less than seconds to minutes per frame respectively. The biggest system we timed was the 83 million particle model of a mitochondrion 10, which took over 4 hours for a single frame. The system of the mitochondrion was the only system which could not be handled with 16GB of RAM and needed a specialized node to run. Its memory consumption peaked at 150GB.

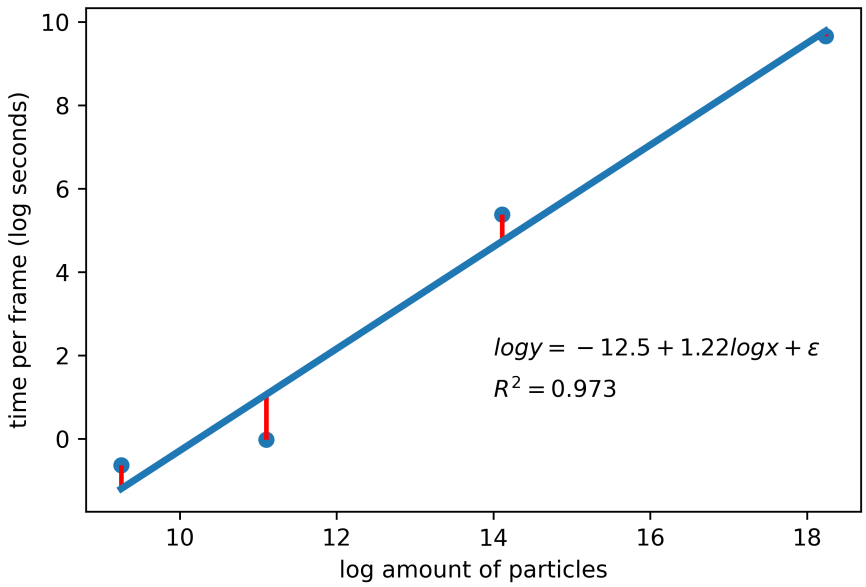

Figure 7: Single frame scalability. The time it takes to segment a single frame with increasing particle count. Exact data is represented in table 1. Some variation from linearity is expected due to the fact that the specific geometry of a system can have an impact on the complexity.

Discussion

Future development of MDVoxelSegmentation

In this manuscript we show the use of voxel based approaches for analysis of lipid and surfactant based systems. However, our implementation is not optimal yet. What we hope to

Table 1: Performance of MDVoxelSegmentation. All systems were run on the same Ubuntu desktop except for the mitochondrion. The test machine contained a Ryzen 1600 (6 cores, hyper threading 3.8 GHz) CPU and 16GB of DDR4 memory (2x 8GB 2.133 MHz). No GPU acceleration is currently implemented. The MDVoxelSegmentation parameters were the same as stated in the examples above except for force-segmentation, which was turned off (-fs 0). Timing was performed using the default Ubuntu time functionality. The CPU for the mitochondrion system ran at 2.4 GHz for a single thread.

System	Particles	Frames	Threads	Real time	User time	Sec/Frame
Bilayer pore	10,436	1,500	12	2m25s	21m41s	0.096
Fatty acid growth	66,278	1,000	12	3m35s	27m37s	0.215
Bilayer pore	10,436	1,500	1	13m13s	15m23s	0.529
Fatty acid growth	66,278	1,000	1	16m20s	18m30s	0.980
Plasma membrane	1,343,450	1	1	4m35s	4m36s	275
Mitochondrion	83,288,300	1	1	260m18s	-	15,618

achieve is to stimulate developers of MD software which have a large user base (MDAnalysis,[12] VMD,[15] Pymol,[23] UnityMol[24]) to embrace voxels and add user intuitive voxel APIs with common components analysis to their packages. VMD already has a well optimized grid density function, which could be made more accessible for users, or even contain some GUI. The VMD code could probably perform all operations in this manuscript at runtime for systems of considerable size due to their efficient usage of GPU(s). How the segmentation API could best be merged into the current selection syntax is not trivial, and will require careful consideration.

Potential extra features

As shown here, leaflet segmentation can be solved with high fidelity, using only local checks, and chronological segmentation. We showed accurate segmentation of a wide range of lipid systems with or without the presence of cholesterol, protein and or (extreme) curvature. Although the focus of this manuscript was lipid leaflet detection, we would like to point at some possible additional features that further broaden the scope of the applications. One of those features is the tracking of membrane pores and stalks. This should use the same basic voxel operations such as erosion, growth and common neighbor segmentation, but results in

tracking abstract segments such as pores. This would allow us to treat pores as segments, propagating them through the trajectory using the same selection syntax and data structures. Current attempts show to be promising, although this pore segmentation appears to rely on a good understanding of subdividing the segmentation results. Some preliminary attempts of the porefinding can be found in the MDVoxelSegmentation github pore branches.

Another possible application is in the analysis of interfaces and the manner in which they percolate between set targets. In this case, the interphase itself might be the entity tracked over time using a distance query on multiple segments.

Detecting lipid lateral phase separation is another problem which we think could be potentially tackled with our tool. In this case we should not map lipid densities to voxels, but lipid compositions. In the voxeled compositions a 3D edge detection algorithm could be used to find transitions in local composition. Since we know that a phase its composition should be constant in the bulk region, we can start with finding volumes which are constant in composition by taking the derivative. This would work similar to the work of Sodt et al. where they demonstrate such approach on a 2D grid.[25] Probably some form of compositional smoothing or dimensionality reduction would be required to reduce noise in the spatial composition of complex systems.

Finally, the analysis is generic enough to be used practically without adaptations on lipid related systems such as amphipathic proteins or fatty acids, as seen in the SI videos 5 and 6, thus showing promise to be used as a more general procedure for amphipathic systems.

To conclude

We showed that consistent and accurate leaflet segmentation can be obtained with our sequential segmentation algorithm. The algorithm proved to be successful on a wide range of lipid and lipid related systems. These systems contained both high compositional and topological complexity which were not segmentable with any of the currently available methods known to us. To achieve this we used a voxel data structure, which allows for fast local

queries, segmenting each frame individually. To achieve consistent segmentation over multiple frames, this was followed by set theory, making use of the Jaccard index. In our tests we show many examples of CG Martini lipid systems, mostly performed with the same parameter settings. These settings show very few artefacts indicating that in many cases no time needs to be spent on parameter tweaking. Besides CG systems we also looked at an atomistic system for which segmentation showed to be of high quality with only minor changes needed to the default settings.

Our algorithm practically scales linearly with the increase in particles and is able to handle millions of particles in minutes on a desktop machine. However, performance gains of orders of magnitude could probably be achieved by proper SIMD/GPU implementation[26] and careful memory management using more advanced voxel data structures16–18. The resulting segmentation is easy to work with in python or VMD and can directly be used for visualization due to its supported segmentation data format.

The MDVoxelSegmentation package will be expanded by our own attempts to tackle problems within this framework. We do invite interested researchers to contribute to the further development of the tool. To add your own segmentation routines to the package, simply open a ticket at our github and indicate your interest.

(https://github.com/marrink-lab/MDVoxelSegmentation)

Methods

Molecular dynamics simulations

To simulate the lipid only systems we made use of GROMACS 2019.1.[27] Martini 2 was used as the force field.[21] All lipid parameters were taken from cgmartini.nl and initial configurations were generated using insane.[20] Each simulation contained 0.15 M NaCl and 10% antifreeze (WF) to prevent unnatural freezing of the water as described in the original Martini paper.[21] Rectangular periodic boundary conditions were employed to prevent artificial wall effects

(xy 10 nm). The steepest descent algorithm was used for energy minimization (1,000 steps) and equilibration was performed using the default Martini settings with a time step of 2 fs for 25,000 steps.[21,28] The Verlet cut-off scheme was used with a 1.1 cut-off for both the coulombic (reaction-field) and van der Waals interactions. The v-rescale scheme (tau-p=1 ps) was used for the thermostat at 310 K, coupling the lipids and solvent/ions in two separate groups. Pressure coupling during equilibration was performed using the Berendsen barostat for semi-anisotropic systems (tau-p=3 ps, compressibility=3e-4 bar-1, ref-p=1.0 bar in each dimension).[29] The production run made use of a 20 fs time step and the pressure coupling was switched to Parrinello-Rahman (tau-p=12).[30]

To simulate pores in the DLPC bilayer, positional restraints were used in the form of a cylindrical biasing potential acting on the C1A and C1B tail beads (GROMACS potential function 2, shape 8, hole radius -1.2 nm, force constant 1,000 kJ mol-1). The negative sign in front of the hole radius indicates an inversion of the biasing potential in GROMACS. A reference file was created with all beads at the same position, acting as the pore center reference position (-r flag in GROMACS). The pore formation was simulated by running a production run for 10 ns without the biasing potential, then the potential was turned on for another 10 ns and turned off again for the final 10 ns.

Leaflet segmentation

We used the same default settings for all the presented segmentation unless specified differently. These settings used a binning resolution of 5 Å and made use of hyper-resolution of 0.5 (adding points half the resolution away from the source point). The headgroups, linkers, tails and exclusions were as specified in the default selection input. The minimum size of a segment was 50 particles and iterative force-segmentation was turned on with a maximum distance of 20 Å. An identity threshold of 0.618 was used for the Jaccard threshold. All keyword arguments represent default values which do not need to be specified except for the reference frame and trajectory file (any MDAnalysis supported MD format).

```
mdvseg -f start.gro -x trajectory.xtc
  --headgroups martini_heads --linkergroup martini_linkers
  --tailgroups martini_tails --exclusiongroup martini_proteins
  --resolution 5 --hyper_resolution 0.5
  --recursion_depth 10 --force_segmentation 20
  --force_information False --minimum_size 50 --begin 0
  --end None --stride 1 --threads 12 --bit_size 32
  --output clusters --verbose False
```

For the segmentation of the atomistic H_{II} system hyper-resolution was turned off (–force_segmentation 0). In general hyper-resolution is not required for atomistic systems. The respective headgroups, tails and linkers selections were specified in the selections input file:

```
[charmm_heads]
"name N P C12 C11 O11 O12 O13 O14"
[charmm_linkers]
"name C1 C2 O21 C21 C3 O31 C31"
[charmm_tails]
"name C22 C23 C24 C25 C26 C27 C28 C29 C210 C211 C212 C213
  C214 C215 C216 C217 C218 C32 C33 C34 C35 C36 C37 C38 C39
  C310 C311 C312 C313 C314 C315 C316 C317 C318"
```

The AA terminal input was:

```
mdvseg -f start.gro -x trajectory.xtc -hg charmm_heads
  -lg charmm_linkers -tg charmm_tails -hres 0
```

Lipid flip-flop

The average z-height of the cholesterol headgroups was subtracted from the headgroup z-position. This resulted in headgroups lying either above or below zero. Headgroups with a normalized z-position less than 0.6 nm away from 0 were labeled either up or down (discretization of the z-height). Differentiating the individual lipid traces results in peaks at flip-flop events. These peaks were counted, registering if the flip occurred up- or downwards. This calculation was repeated for all cholesterols in all frames and a graph was made using python3 matplotlib.[31] The z-height and flip-flop analysis is available at the MDVoxelSegmentation github.

To show the flip-flopping of cholesterol using our leaflet segmentation we used the default settings except for the force-segmentation which was turned off. Lipids which were not assigned to segment 0 were assigned to their previous segment. A graph was made using matplotlib and the plotting script included with mdvseg. The leaflet flip-flop analysis is available at the MDVoxelSegmentation github.

Visualization

All figures and videos were rendered using VMD and a little TCL script of MDVoxelSegmentation to load the segmentation array. Examples of TCL scripts are provided together with the code at:

https://github.com/marrink-lab/MDVoxelSegmentation

Acknowledgement

Bart M. H. Bruininks thanks Manuel N. Melo, Jonathan Barnoud, Weria Pezeshkian, John Stone and the rest of the VMD developers, as well as Elmar Eisemann for the many fruitful discussions. We also thank Melanie König (fatty acids), Haleh Abdizadeh (plasma membrane), Josef Melcr (amphipathic peptides), Weria Pezeshkian (mitochondrion) and Peter

Tieleman (atomistic H_{II}) for sharing their data.

Supporting Information Available

Supporting Figures

Comparing leaflet detection algorithms

A qualitative comparison of segmentation quality between MDAleaflets,[12] FATSLiM[14] and MDVoxelSegmentation was performed to explore the strengths and weaknesses of each approach.

Program	Temporal Consistency	Monolayer	Stacked Bilayers	Bilayer with Cholesterol	Speed Frame*
FATSLiM	no	no	yes	yes	20
MDAleaflets	no	yes	no	not safe**	4
MDVseg	yes	yes	yes	yes	1

Program	Bilayer with Protein	Bilayer with Toroidal Pore	Inverted Hexagonal	Plasma Membrane	Speed Trajectory*
FATSLiM	yes	no	very unsafe	yes	5
MDAleaflets	not safe**	yes	yes	out of mem (>70GB)***	-
MDVseg	yes	yes	yes	yes	1

Figure 8: Qualitative comparison of leaflet segmentation algorithms. A variety of lipid systems was segmented using the default options of MDAleaflets, FATSLiM and MDVoxelSegmentation (MDVseg). Solid images indicate reliable output. Faded images represent the best case output for unreliable segmentation. The headgroups were selected using the default MDVoxelSegmentation input selection. The large stop sign indicates no output was generated at all during segmentation. Unreliable segmentation is indicated with the fingers crossed symbol. *The speed is indicated relative to the MDVoxelSegmentation performance. **Segmentation can be extremely sensitive to diving lipids. ***The option for sparse matrices does exist, however, it is not the default behavior and the calculation speed suffered severely.

Flip-flop comparison with smaller deadzone

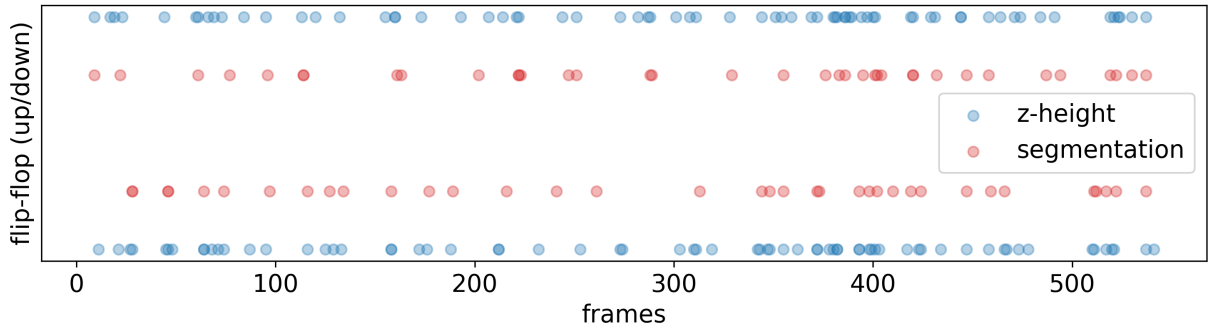

Figure 9: Comparing flip-flop detection with a small deadzone. Reducing the deadzone for the z-height based flip-flop registration to 0.3 showed that the flip-flops detected by the MDVoxelSegmentation approach is a near perfect subset of the z-height approach with a small deadzone. Although the exact moment of the flip-flop is sometimes shifted by a few frames between the two approaches.

The role of the Jaccard threshold

For temporal segmentation a Jaccard threshold had to be set. This threshold defines when two sets are similar enough to be a candidate for shared identity. Picking a very low threshold results in an overzealous attempt to never generate a new set, where a very high threshold results in a flood of new identities – a single lipid flip-flopping can cause a loss of identity. We found that values around the golden ratio correspond well with our human intuition.

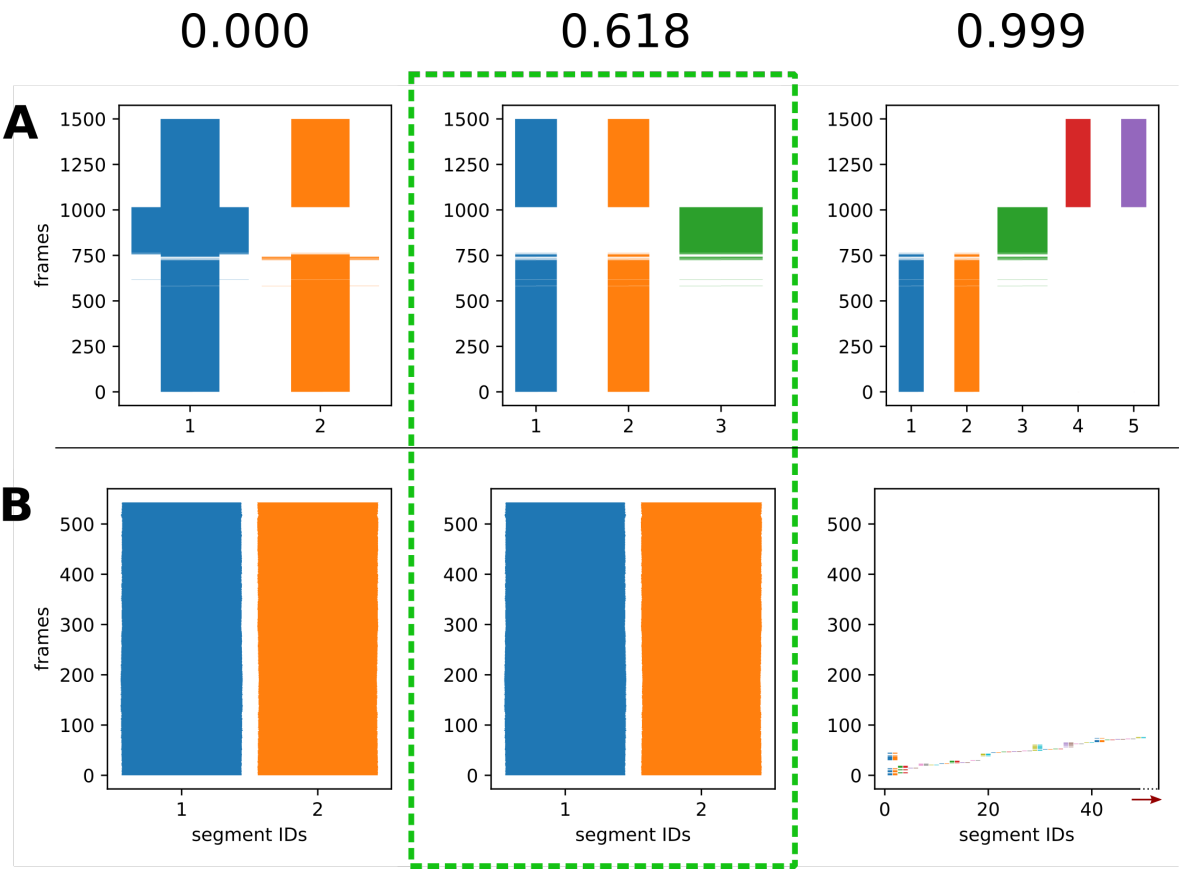

Figure 10: Comparing Jaccard thresholds. Three values for the Jaccard threshold (0.000, 0.618, 0.999) were tested on two bilayer systems. A bilayer with a toroidal pore (A). A bilayer with cholesterol (B). The Jaccard threshold of 0.618 gives the desired results in both systems. Setting the threshold to 0 results in incorrect temporal segmentation of system A, although system B is fine. Setting the threshold to 0.999 results in bad temporal segmentation in both systems. Segment IDs beyond 50 were truncated.

Supporting Videos

Supporting Video 1: Bilayer pore

bilayer_pore The opening and closing of an artificial pore in a DLPC bilayer. Right before the stable toroidal pore is formed, a short lived water pore spans the bilayer. This pore was visualized using the following selection syntax in VMD: '(name W) and (within 20 of name PO4 and user 1) and (within 20 name PO4 and user 2)'. User 1 and 2 are the respective leaflets. We can visualize water pores connecting two leaflets and segment them. However,

we lose all information of the pore as soon as the leaflets become continuous (as illustrated in the video).

Supporting Video 2: Protein bilayer

protein_bilayer Stable segmentation of a bilayer in the presence of a protein. Close to the protein some lipids can be observed which are diving relatively deep under the surface, however, this diving is handled without missegmentation.

Supporting Video 3: Lipid phase transition

phase_transition Lipid segmentation of a phase transition from dehydrated stacked bilayers to inverted hexagonal (left). Water densities (right).

Supporting Video 4: Lipoplex transfection

transfection A lipoplex was placed on top of a model endosomal membrane. The simulation was started in the presence of a fusion stalk, connecting the outer leaflet of the lipoplex coat with the upper leaflet of the endosomal membrane (red, transparent). Over time the internal compartments of the lipoplex (orange, yellow, pink, green) fuse with the bottom endosomal leaflet (red, transparent). At the end of the simulation all dsDNA from inside the lipoplex was transfected over the endosomal membrane. Some flickering is observed, but overall the segmentation is very stable and flickering could easily be removed with a noise filter.

Supporting Video 5: Acyl chain dinner

acyl_chains, acyl_chains_dof In an attempt to understand early life 'cell' division we placed a small vesicle (acyl chains) in the presence of food (also acyl chains). The vesicle adds the food from its surroundings to its own composition, but without internal reorganization the food is only added to the outer leaflet. With a large enough leaflet disbalance three things

can happen to restore the balance. The acyl chains flip-flop from the outside to the inside. The leaflets temporarily fuse to balance the amount of acyl chains in the inner and outer leaflet, or the disbalance causes the vesicle to divide. MDVoxelSegmentation was used to analyse the imbalance as well as to monitor the feeding rate. Both flip-flopping and toroidal pore formation takes place. No divisions occur.

Supporting Video 6: Segmentation of other amphipathic systems

amphipathic_peptides With trivial adaptations of the input GRO file we used our method to perform segmentation of amphipathic peptides. Due to their amphipathic nature, micelles from such peptides can touch but not fuse (like lipid leaflets). MDVoxelSegmentation shows to be suited to be used for such systems.

References

(1) McCammon, J. A.; Gelin, B. R.; Karplus, M. Dynamics of folded proteins. *Nature* **1977**, *267*, 585–590.

(2) Marrink, S. J.; Corradi, V.; de Souza, P. C.; Ingólfsson, H. I.; Tieleman, D. P.; Sansom, M. S. Computational Modeling of Realistic Cell Membranes. *Chemical Reviews* **2019**, *119*, 6184–6226.

(3) van Eerden, F. J.; de Jong, D. H.; de Vries, A. H.; Wassenaar, T. A.; Marrink, S. J. Characterization of thylakoid lipid membranes from cyanobacteria and higher plants by molecular dynamics simulations. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **2015**, *1848*, 1319–1330.

(4) Bruininks, B. M. H.; de Souza, P. C. T.; Ingolfsson, H.; Marrink, S. J. A molecular view on the escape of lipoplexed DNA from the endosome. *eLife* **2020**, *9*.

(5) Vögele, M.; Bhaskara, R. M.; Mulvihill, E.; van Pee, K.; Özkan Yildiz,; Kühlbrandt, W.; Müller, D. J.; Hummer, G. Membrane perforation by the pore-forming toxin pneumolysin. *Proceedings of the National Academy of Sciences* **2019**, *116*, 13352–13357.

(6) Ingolfsson, H. I.; Tieleman, P.; Marrink, S. Lipid Organization of the Plasma Membrane. *Biophysical Journal* **2015**, *108*, 358a.

(7) Perilla, J. R.; Hadden, J. A.; Goh, B. C.; Mayne, C. G.; Schulten, K. All-Atom Molecular Dynamics of Virus Capsids as Drug Targets. *The Journal of Physical Chemistry Letters* **2016**, *7*, 1836–1844.

(8) Ramezanpour, M.; Schmidt, M. L.; Bashe, B. Y. M.; Pruim, J. R.; Link, M. L.; Cullis, P. R.; Harper, P. E.; Thewalt, J. L.; Tieleman, D. P. Structural Properties of Inverted Hexagonal Phase: A Hybrid Computational and Experimental Approach. *Langmuir* **2020**, *36*, 6668–6680.

(9) Holdbrook, D. A.; Huber, R. G.; Piggot, T. J.; Bond, P. J.; Khalid, S. Dynamics of Crowded Vesicles: Local and Global Responses to Membrane Composition. *PLOS ONE* **2016**, *11*, e0156963.

(10) Pezeshkian, W.; König, M.; Wassenaar, T. A.; Marrink, S. J. Backmapping triangulated surfaces to coarse-grained membrane models. *Nature Communications* **2020**, *11*.

(11) Alasiri, H. Determining Critical Micelle Concentrations of Surfactants Based on Viscosity Calculations from Coarse-Grained Molecular Dynamics Simulations. *Energy & Fuels* **2019**, *33*, 2408–2412.

(12) Gowers, R.; Linke, M.; Barnoud, J.; Reddy, T.; Melo, M.; Seyler, S.; Domański, J.; Dotson, D.; Buchoux, S.; Kenney, I.; Beckstein, O. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. Proceedings of the 15th Python in Science Conference. 2016.

(13) Sega, M.; Hantal, G.; Fábián, B.; Jedlovszky, P. Pytim: A python package for the interfacial analysis of molecular simulations. *Journal of Computational Chemistry* **2018**, *39*, 2118–2125.

(14) Buchoux, S. FATSLiM: a fast and robust software to analyze MD simulations of membranes. *Bioinformatics* **2016**, *33*, 133–134.

(15) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *Journal of Molecular Graphics* **1996**, *14*, 33–38.

(16) Dado, B.; Kol, T. R.; Bauszat, P.; Thiery, J.-M.; Eisemann, E. Geometry and Attribute Compression for Voxel Scenes. *Computer Graphics Forum* **2016**, *35*, 397–407.

(17) Villanueva, A. J.; Marton, F.; Gobbetti, E. SSVDAGs. Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. 2016.

(18) Careil, V.; Billeter, M.; Eisemann, E. Interactively Modifying Compressed Sparse Voxel Representations. *Computer Graphics Forum* **2020**, *39*, 111–119.

(19) Sualeh, M.; Kim, G.-W. Dynamic Multi-LiDAR Based Multiple Object Detection and Tracking. *Sensors* **2019**, *19*, 1474.

(20) Wassenaar, T. A.; Ingólfsson, H. I.; Böckmann, R. A.; Tieleman, D. P.; Marrink, S. J. Computational Lipidomics with insane: A Versatile Tool for Generating Custom Membranes for Molecular Simulations. *Journal of Chemical Theory and Computation* **2015**, *11*, 2144–2155.

(21) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *The Journal of Physical Chemistry B* **2007**, *111*, 7812–7824.

(22) Bruininks, B. M. H.; de Souza, P. C. T.; Marrink, S. J. *Methods in Molecular Biology*; Springer New York, 2019; pp 105–127.

(23) LLC, S. The PyMOL Molecular Graphics System. 2020; `https://pymol.org/2/`.

(24) Doutreligne, S.; Cragnolini, T.; Pasquali, S.; Derreumaux, P.; Baaden, M. UnityMol: Interactive scientific visualization for integrative biology. 2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV). 2014.

(25) Sodt, A. J.; Sandar, M. L.; Gawrisch, K.; Pastor, R. W.; Lyman, E. The Molecular Structure of the Liquid-Ordered Phase of Lipid Bilayers. *Journal of the American Chemical Society* **2014**, *136*, 725–732.

(26) Cabaret, L.; Lacassagne, L.; Oudni, L. A review of world's fastest connected component labeling algorithms: Speed and energy estimation. Proceedings of the 2014 Conference on Design and Architectures for Signal and Image Processing. 2014.

(27) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

(28) de Jong, D. H.; Baoukina, S.; Ingólfsson, H. I.; Marrink, S. J. Martini straight: Boosting performance using a shorter cutoff and GPUs. *Computer Physics Communications* **2016**, *199*, 1–7.

(29) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics* **1984**, *81*, 3684–3690.

(30) Parrinello, M.; Rahman, A. Polymorphic transitions in single crystals: A new molecular dynamics method. *Journal of Applied Physics* **1981**, *52*, 7182–7190.

(31) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **2007**, *9*, 90–95.

Graphical TOC Entry

Some journals require a graphical entry for the Table of Contents. This should be laid out "print ready" so that the sizing of the text is correct.
Inside the `tocentry` environment, the font used is Helvetica 8 pt, as required by *Journal of the American Chemical Society*.
The surrounding frame is 9 cm by 3.5 cm, which is the maximum permitted for *Journal of the American Chemical Society* graphical table of content entries. The box will not resize if the content is too big: instead it will overflow the edge of the box.
This box and the associated title will always be printed on a separate page at the end of the document.