JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

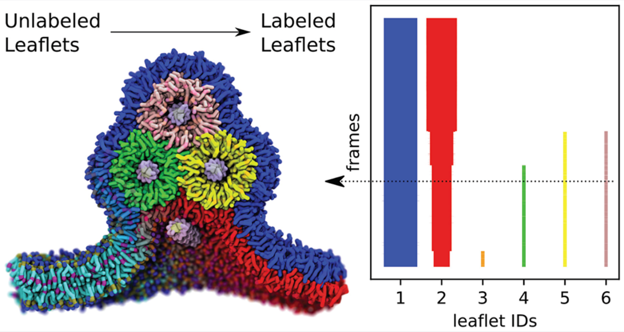
Sequential Voxel-Based Leaflet Segmentation of Complex Lipid Morphologies

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

Cite This: *J. Chem. Theory Comput.* 2021, 17, 7873−7885 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

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, because of their amphipathic nature, becoming more intricate as the particle count increases. A few lipids might form a micelle, where aggregation of tens of thousands could lead to vesicle formation. Millions of lipids comprise 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 that understand what is meant by emerging entities such as micelles and vesicles. Studying such systems at the particle level only becomes extremely tedious, counterintuitive, 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 mesoscale. By using a voxel-based approach and focusing on locality, we forego 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 prefitting parameters, and segmentation of millions of particles can be achieved on a desktop machine.

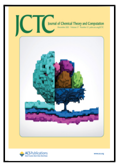


■ INTRODUCTION

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 they may fail as systems contain more than one or two segments (see Figure S1 in the Supporting Information). 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

Received: May 6, 2021
Published: October 5, 2021



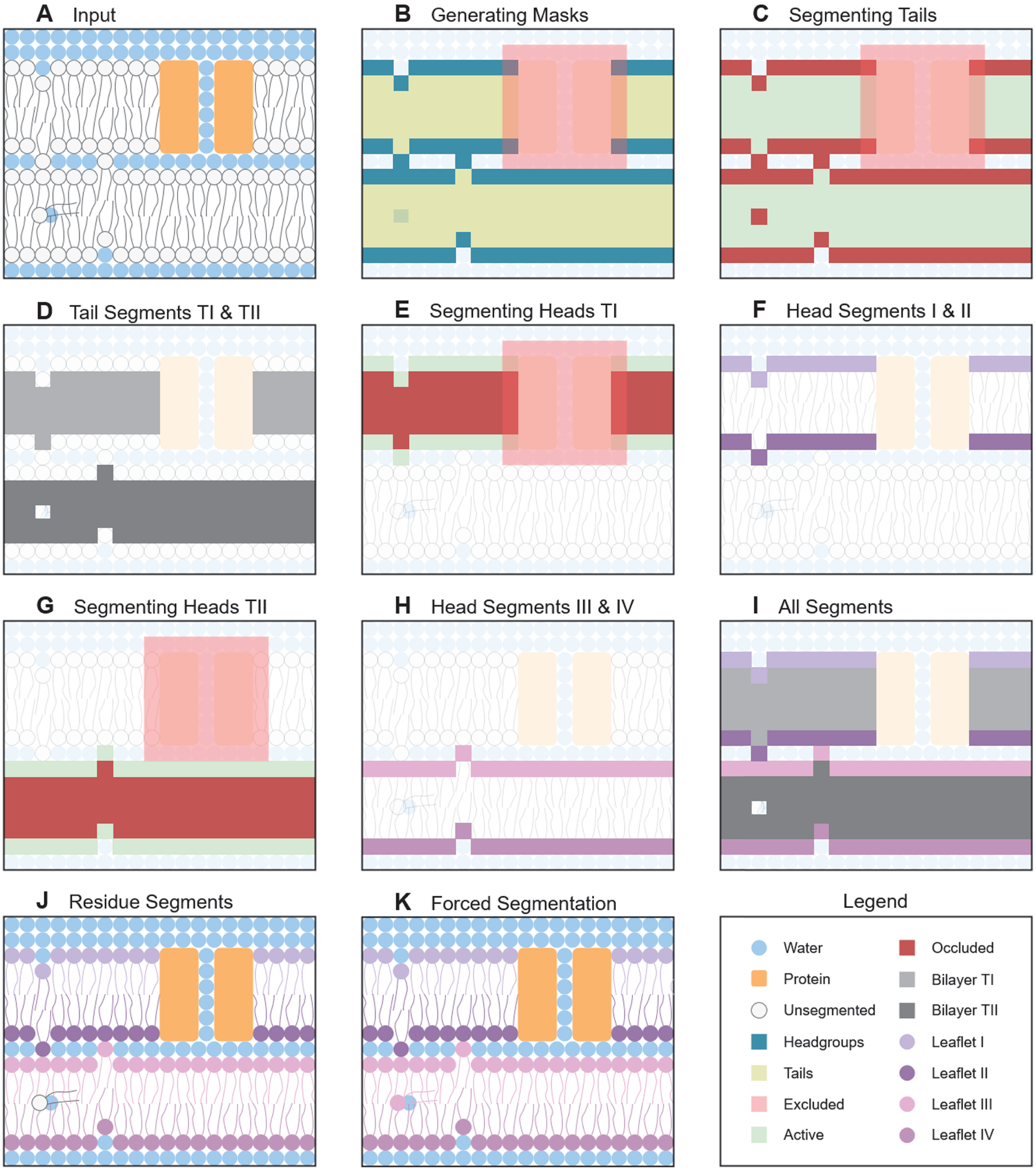


Figure 1. Spatial leaflet segmentation. A 2D schematic representation of the leaflet segmentation of two stacked bilayers in a periodic box. The upper bilayer contains a protein channel and the lower bilayer contains a diving lipid. The protein is used as exclusion selection.

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 SI files contain 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 multistep connective components analysis for lipid-based systems. We then 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 file, a trajectory file, 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]

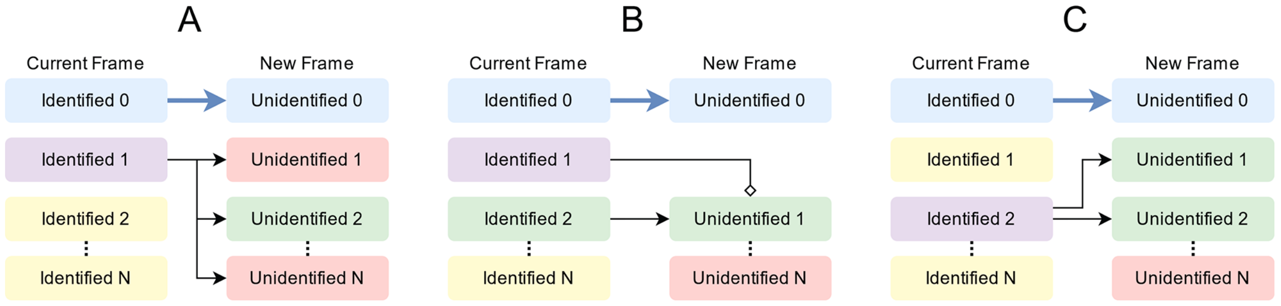


Figure 2. Finding the identity of the segments in the new frame. (A) The ordered 1 segment has a best match with the unordered 2 segment. (B) The ordered 1 and 2 segments are merged into one segment, with the ordered 2 segment providing the identity for the unordered 1 segment. (C) The ordered 2 segment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

meaning no selection is required 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 Figure 1.

Leaflet segmentation starts with reading the input files (Figure 1A) and mapping all the lipid tails, lipid heads, and exclusion beads to boolean voxel matrices (0.5 nm binning for Martini lipid systems; see Figure 1B). 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 (Figure 1C). 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 Figure 1D.

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 (see Figures 1E and 1G). This is achieved by using only those headgroups which have a lipid residue index that 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 (Figure 1F and 1H).

Although the result at this point is good (Figures 1I and 1J), there might be some lipids that still have no segment assigned. Force segmentation can solve this problem, because it can be used to map lipids to a segment based on a distance criterion. For every lipid that does not have a segment assigned, all neighboring lipids are checked within a certain cutoff distance (default is 2 nm). If the dominant segment in that search is not 0 (unsegmented), the lipid is assigned 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 Figure 1. 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, whereas 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, while, at the same time, hardly affecting the segmentation quality.

The result after spatial leaflet segmentation of a trajectory is a new trajectory that contains a segmentation entry for each index in each frame. These indexes correspond to the atom indices in the original structure file. Atoms that have no segmentation are assigned to 0. However, because of the arbitrary picking of our initial voxels for segmentation, the segmentation labels are missing sensible consistency over time for all other labels than 0. To solve this lack of chronological consistency, we must 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 must 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 (Figure 2A). 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, because of shrinkage 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 $M1$

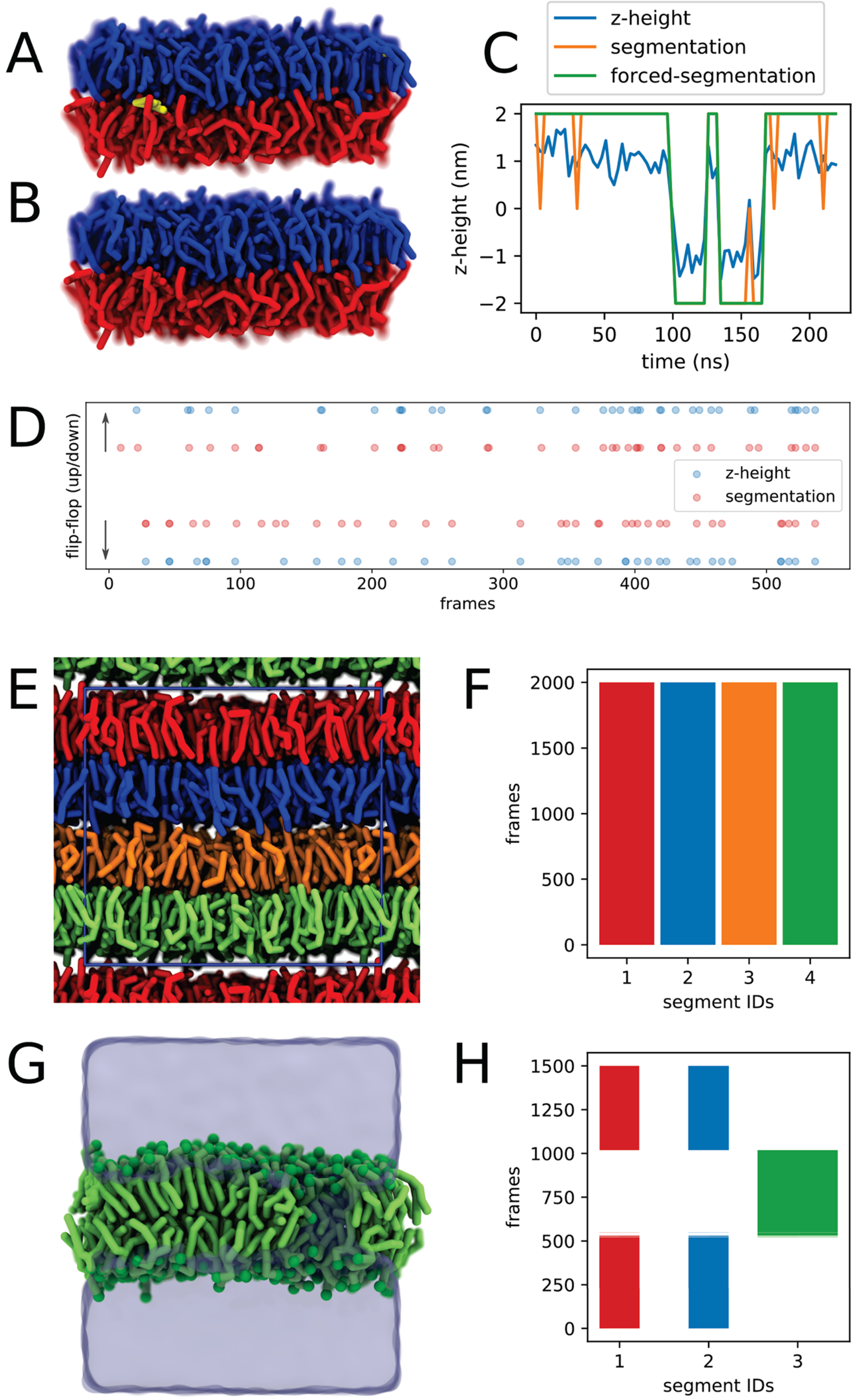


Figure 3. Leaflet segmentation and analysis for pure bilayer systems. (A, B) Snapshots of a DOPC/CHOL(4:1) bilayer with and without force segmentation. Some lipids were unassigned without force segmentation, because of diving/occlusion (panel (A), yellow). Using force segmentation, nonsegmented lipids are assigned to a segment as a postprocessing step (panel (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) Comparison of the upward 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) 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) Snapshot of a DLPC bilayer with a toroidal pore (frame 750 of panel (H)). (H) Leaflet segmentation of pore formation; the pore-forming biasing potential was turned on between frame 500 and 1000, and the width of a segment represents its relative size.

and *N*2 to *M*2, to allow for the growing and shrinking of segments. Therefore, we need a way to identify the amount of shared elements between two segments.

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

https://doi.org/10.1021/acs.jctc.1c00446

To identify how many atoms are shared between two segments across frames the Jaccard index is used (see 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 (see Figure S2 in the Supporting Information). 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 that has lost lipids because of flip-flopping can still be identified 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. Therefore, it 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 been 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.

In the case of a merge (Figure 2B), 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 that 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, since, otherwise, the identity would be inherited. This information can be used to restore identities when splitting events occur.

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 (Figure 2C). 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 losing 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 that 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. A 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 (Figure 3). We then add proteins and a high variety of compositional complexity (Figure 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 (Figure 5). We end with some more general real-world examples, including an example at atomistic resolution (Figure 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 (see orange components in Figures 3A and 3C). By turning on force segmentation, all lipids including the cholesterol could be assigned successfully to a leaflet (see green components in Figures 3B and 3C).

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 that 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 (Figure 3C). 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 (Figure 3D). 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, almost all flip-flops indicated by the segmentation analysis are in the set of the *z*-height approach (see Figure S3 in the Supporting Information). In other words, it appears that the flip-flops indicated by the segmentation are an almost-perfect subset of the *z*-height approach with a relatively small deadzone.

Since stacked bilayers are notoriously difficult 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, Figure 3E). The close proximity and or intercalation of the bilayers did not hinder correct leaflet segmentation at all over a trajectory spanning 10 μs (Figure 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 (see Figures 3G and 3H, as well as Video 1 in the Supporting Information).

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.

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 (see Figure 4A, as well as Video 2 in the SI). Next, we tested a crowded plasma membrane model. This model contains a high variety of lipids (~60) and a wide range of proteins in a planar membrane.[6] Segmentation resulted in an almost-perfect assignment (Figure 4B). However, there are 12 lipids that remain unlabeled (see the highlighted box in Figure 4B), 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. Until 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 that should push the time consistency part of our algorithm, as well as the agonistic behavior, with respect to the amount of segments. A snapshot of the first intermediate and last frame of the trajectory is shown in Figure 5A. 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 that share a common water reservoir (Figure 5A, panel I) fuse together by multiple stalk formation (Figure 5A, panel II). The leaflets are then split into smaller segments, more closely resembling the classic view of the H_{II} phase. However, because of 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 (Figure 5A, panel 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 Methods section) immediately makes it clear which channels are connected and which are not.

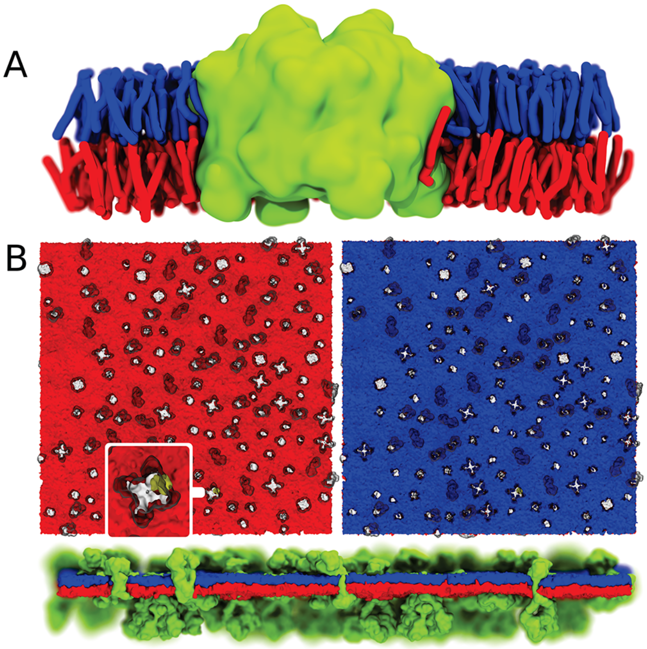


Figure 4. Leaflet segmentation in the presence of protein. (A) Adding protein (lime) to a bilayer does not interfere with the leaflet assignment of a CG POPC bilayer (red/blue). A video of the bilayer protein system can be found in the SI. (B) The bottom and top leaflet of the plasma membrane. Even extremely 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 nonassigned lipids (yellow) around one of the proteins (highlighted box).

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 analyze. A rendering of the leaflet segmentation is shown in Figure 5B. 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

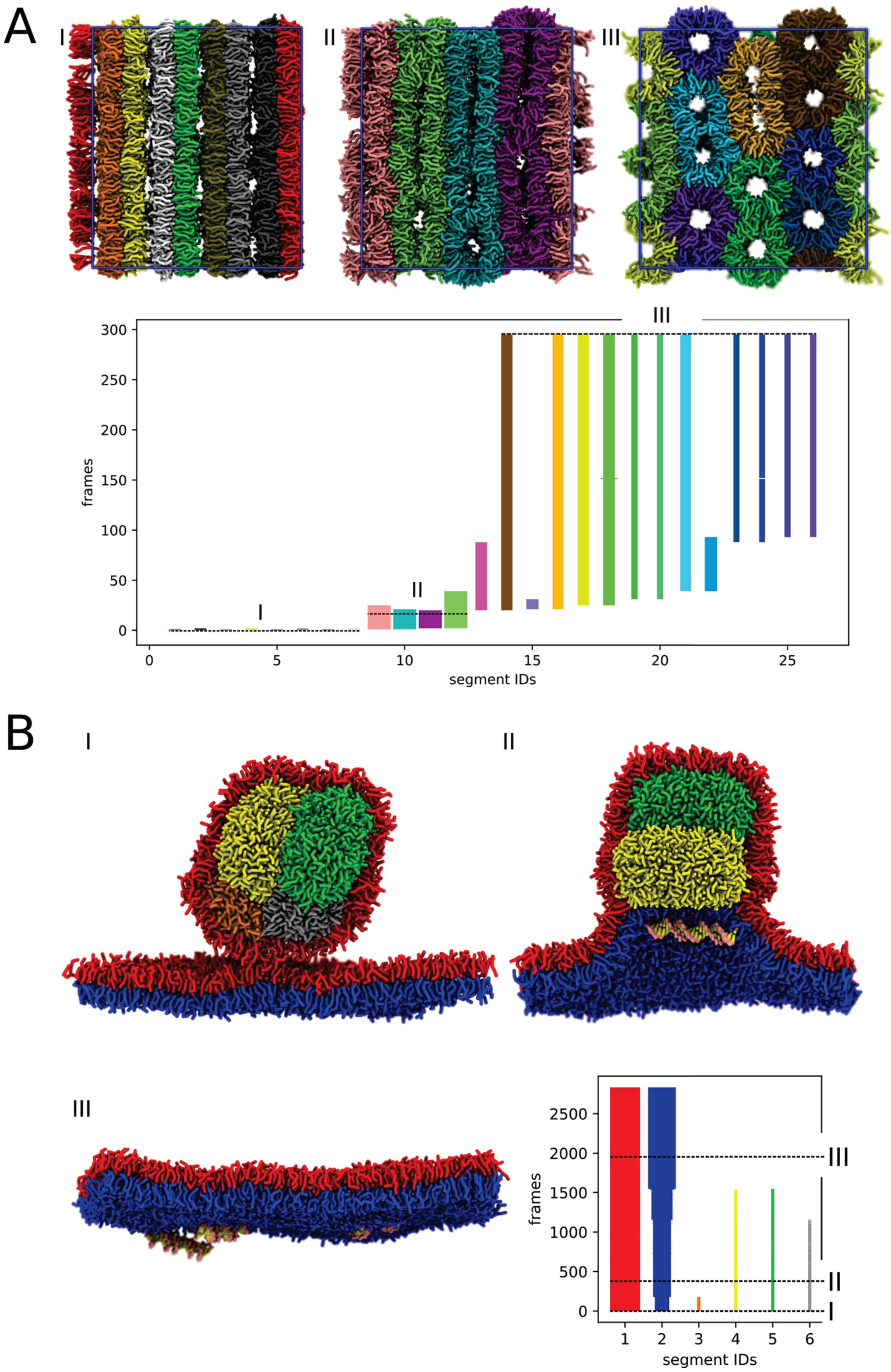


Figure 5. Leaflet segmentation and analysis for both tight and curved complex geometry. (A) The phase transition from a multilamellar bilayer system to an H_{II} phase. Snapshots of the first, intermediate and last segmented frame of a 1 μs trajectory. The accessory video is available in the SI as Video 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.

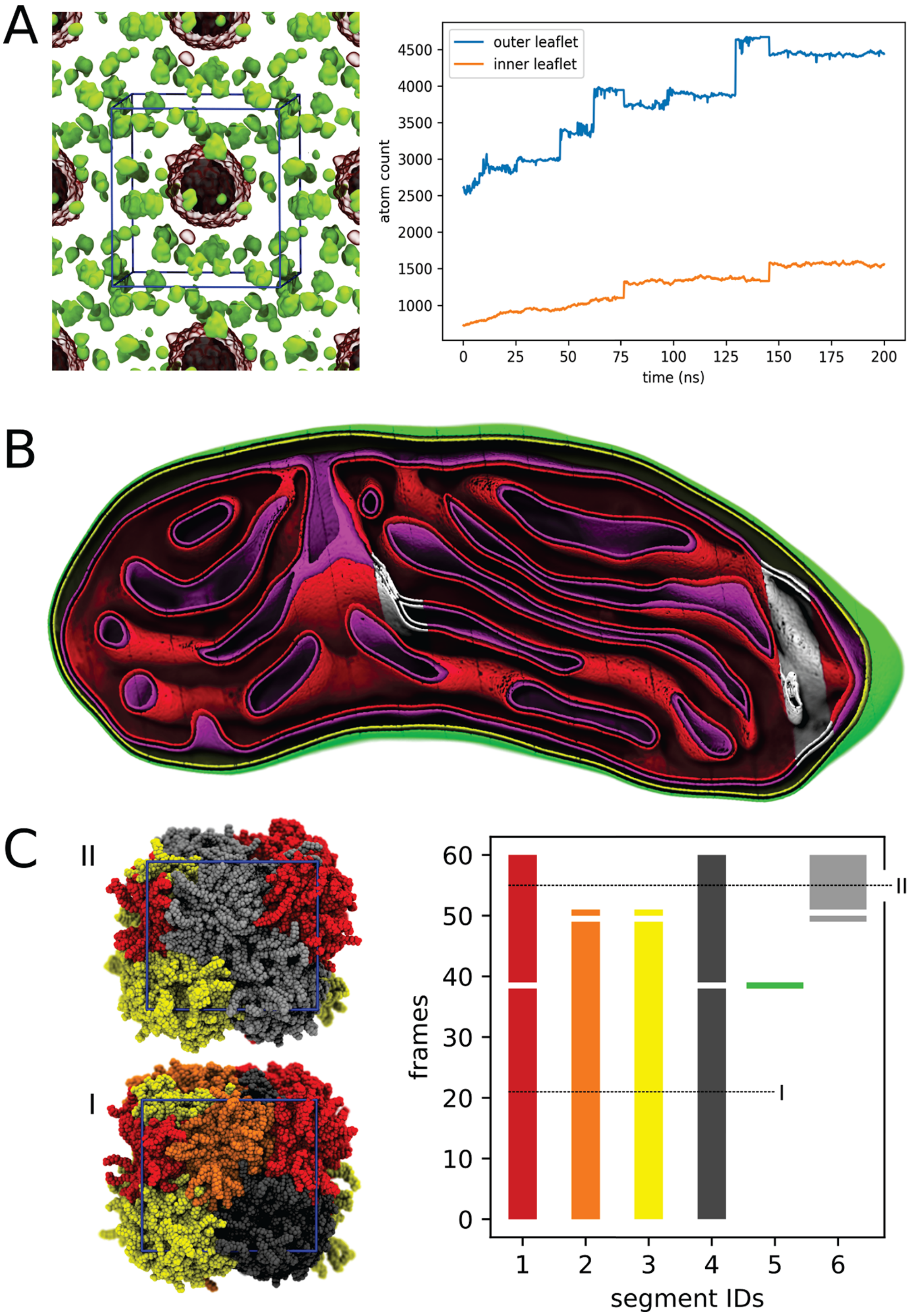


Figure 6. Additional systems. (A) Snapshot of a fatty acid vesicle (red/gray), surrounded by individual fatty acids in the medium (green); the amount of fatty acids in the inner and outer leaflet of the vesicle over time is shown as a graph, and a video of the fatty acid dimer is available in the SI Video 5. (B) 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.

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 (Figure 5B, graph). We did find that decreasing the resolution of the leaflet segmentation for complex geometry resulted in a steep decrease in the 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 are 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.

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 (see Figure 6, as well as Video 5 in the SI). 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, because of 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 (Figure 6A). 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 20 slices. Each piece was segmented individually (Figure 6B). This revealed that there are issues in two of the 20 slices. When these two problematic slices are excluded, the expected four leaflets were found by segmentation analysis. This indicates that 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 (see Figure 5A).[4] These channels probably have huge effects on the fusion kinetics of such phases. Related work was performed on an 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 middle to end of the trajectory of their lipid system with the highest hydration (Figure 6C). 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, since 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 re-evaluating 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 that are otherwise easily missed or extremely cumbersome to analyze.

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, since it can have a large impact on performance based on the exact system (especially at higher maximum recursion depth). The results are summarized in Figure 7. From the single thread values, we can deduce that our algorithm indeed scales pretty much linearly with some variance, because of packing and complexity (see 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 mitochondrion,[10] which took over 4 h for a single frame. The system of the mitochondrion was the only system that could not be handled with 16GB of RAM and needed a specialized node to run. Its memory consumption peaked at 150GB.

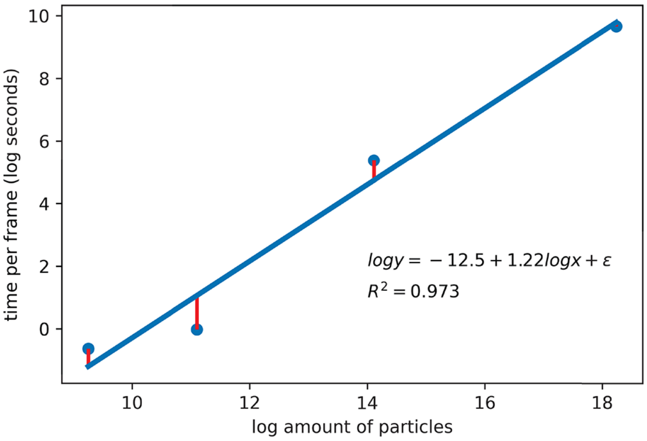


Figure 7. Single frame scalability. The time required to segment a single frame, relative to increasing particle count. Exact data is represented in Table 1. Some variation from linearity is expected due to the fact that the specific geometry of a system can have an impact on the complexity.

DISCUSSION

Future Development of MDVoxelSegmentation. In this manuscript, we show the use of voxel-based approaches for the analysis of lipid- and surfactant-based systems. However, our implementation is not optimal yet. What we hope to achieve is to stimulate developers of MD software who have a

Table 1. Performance of MDVoxelSegmentation[a]

system	number of particles	number of frames	number of threads	real time	user time	sec/frame
bilayer pore	10 436	1500	12	2m, 25s	21m, 41s	0.096
fatty acid growth	66 278	1000	12	3m, 35s	27m, 37s	0.215
bilayer pore	10 436	1500	1	13m, 13s	15m, 23s	0.529
fatty acid growth	66 278	1000	1	16m, 20s	18m, 30s	0.980
plasma membrane	1 343 450	1	1	4m, 35s	4m, 36s	275
mitochondrion	83 288 300	1	1	260m, 18s	–	15 618

[a] All systems were run on the same Ubuntu desktop, except for the mitochondrion. The test machine contained a Ryzen 1600 CPU (6 cores, hyper threading, 3.8 GHz) and 16GB of DDR4 memory (2× 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 operated at 2.4 GHz for a single thread.

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 highlight 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 that we think could be potentially addressed with our tool. In this case, we should not map lipid densities to voxels, but rather lipid compositions. In the voxeled compositions, a three-dimensional (3D) edge detection algorithm could be used to find transitions in local composition. Since we know that, for a phase, its composition should be constant in the bulk region, we can start with finding volumes that are constant in composition by taking the derivative. This would work similar to the work reported by Sodt et al., where they demonstrated such an approach on a two-dimensional (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.

Conclusions. 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 artifacts, indicating that, in many cases, no time is required to be spent on parameter tweaking. Besides CG systems, we also examined 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 within 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 structures.[16−18] The resulting segmentation is easy to work with in the python or VMD programs and can directly be used for visualization, because of its supported segmentation data format.

The MDVoxelSegmentation package will be expanded by our own attempts to address 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 (1000 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 cutoff scheme was used with a 1.1 cutoff 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 1000 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 that 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). Generally, 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 upward or downward. 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 that 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.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.1c00446.

Figures S1−S3; descriptions of Videos S1−S6 (PDF)

Video S1 showing opening and closing of an artificial pore in a DLPC bilayer (MP4)

Video S2 showing stable segmentation of a bilayer in the presence of a protein (MP4)

Video S3 showing lipid segmentation of a phase transition (MP4)

Video S4 showing a lipoplex on top of a model endosomal membrane (MP4)

Video S5 showing a small vesicle placed in the presence of food (MP4)

Video S6 showing method used to perform segmentation of amphipathic peptides (MP4)

■ AUTHOR INFORMATION

Corresponding Author

⊗Bart M. H. Bruininks − *Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, 9712*

CP Groningen, The Netherlands; orcid.org/0000-0001-5136-0864; Email: b.m.h.bruininks@rug.nl

Authors

[⊗]**Albert S. Thie** − *Zernike Institute for Advanced Materials, University of Groningen, 9712 CP Groningen, The Netherlands*

Paulo C. T. Souza − *Molecular Microbiology and Structural Biochemistry (MMSB, UMR 5086), CNRS, University of Lyon, 69367 Lyon, France; Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, 9712 CP Groningen, The Netherlands;* orcid.org/0000-0003-0660-1301

Tsjerk A. Wassenaar − *Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, 9712 CP Groningen, The Netherlands;* orcid.org/0000-0002-6345-0266

Shirin Faraji − *Zernike Institute for Advanced Materials, University of Groningen, 9712 CP Groningen, The Netherlands*

Siewert J. Marrink − *Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, 9712 CP Groningen, The Netherlands;* orcid.org/0000-0001-8423-5277

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.1c00446

Notes

The authors declare no competing financial interest.
[⊗]First authorship is shared between B.M.H.B. and A.S.T.

ACKNOWLEDGMENTS

B.M.H.B. 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.

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.; Souza, P. C. T.; Ingólfsson, H. I.; Tieleman, D. P.; Sansom, M. S. P. Computational Modeling of Realistic Cell Membranes. *Chem. Rev.* **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. *Biochim. Biophys. Acta, Biomembr.* **2015**, *1848*, 1319−1330.

(4) Bruininks, B. M. H.; Souza, P. C. T.; Ingolfsson, H.; Marrink, S. J. A molecular view on the escape of lipoplexed DNA from the endosome. *eLife* **2020**, *9*, e52012.

(5) Vögele, M.; Bhaskara, R. M.; Mulvihill, E.; van Pee, K.; Yildiz, Ö.; Kühlbrandt, W.; Müller, D. J.; Hummer, G. Membrane perforation by the pore-forming toxin pneumolysin. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 13352−13357.

(6) Ingolfsson, H. I.; Tieleman, P.; Marrink, S. Lipid Organization of the Plasma Membrane. *Biophys. J.* **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. *J. Phys. Chem. Lett.* **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*, No. e0156963.

(10) Pezeshkian, W.; König, M.; Wassenaar, T. A.; Marrink, S. J. Backmapping triangulated surfaces to coarse-grained membrane models. *Nat. Commun.* **2020**, *11*, 2296.

(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. In *Proceedings of the 15th Python in Science Conference (SCiPy 2016)*, Austin, TX, July 10−16, 2016; p 98.

(13) Sega, M.; Hantal, G.; Fábián, B.; Jedlovszky, P. Pytim: A python package for the interfacial analysis of molecular simulations. *J. Comput. Chem.* **2018**, *39*, 2118−2125.

(14) Buchoux, S. FATSLiM: a fast and robust software to analyze MD simulations of membranes. *Bioinformatics* **2017**, *33*, 133−134.

(15) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. 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: symmetry-aware sparse voxel DAGs. In *I3D '16: Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Redmond, WA, Feb. 27−28, 2016; pp 7−14.

(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. *J. Chem. Theory Comput.* **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. *J. Phys. Chem. B* **2007**, *111*, 7812−7824.

(22) Bruininks, B. M. H.; Souza, P. C. T.; Marrink, S. J. A Practical View of the Martini Force Field. In *Biomolecular Simulations*; Bonomi, M., Camilloni, C., Eds.; Methods in Molecular Biology, Vol. *2202*; Springer: New York, 2019; pp 105−127.

(23) LLC, S. *The PyMOL Molecular Graphics System*, 2020; available via the Internet at: 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)*, France, Nov. 9−10, 2014; pp 109−110, DOI: 10.1109/LDAV.2014.7013213.

(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. *J. Am. Chem. Soc.* **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. In *Proceedings of the 2014 Conference on Design and Architectures for Signal and Image Processing*, Madrid, Spain, Oct. 8−10, 2014, DOI: 10.1109/DASIP.2014.7115641.

(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. *Comput. Phys. Commun.* **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. *J. Chem. Phys.* **1984**, *81*, 3684−3690.

(30) Parrinello, M.; Rahman, A. Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.* **1981**, *52*, 7182−7190.

(31) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90−95.

https://doi.org/10.1021/acs.jctc.1c00446