

# Enabling Atomistic Modeling and Simulation of Complex Curved Cellular Membranes with xMAS Builder

Noah Trebesch and Emad Tajkhorshid\*

Theoretical and Computational Biophysics Group  
NIH Resource for Macromolecular Modeling and Visualization  
Beckman Institute for Advanced Science and Technology  
University of Illinois Urbana-Champaign

\*Corresponding Author (Email: [emad@illinois.edu](mailto:emad@illinois.edu))

January 14, 2025

## Abstract

As more powerful high performance computing resources are becoming available, there is a new opportunity to bring the unique capabilities of molecular dynamics (MD) simulations to cell-scale systems. Membranes are ubiquitous within cells and are responsible for a diverse set of essential biological functions, but building atomistic models of cell-scale membranes for MD simulations is immensely challenging because of their vast sizes, complex geometries, and complex compositions. To meet this challenge, we have developed xMAS Builder (**EX**perimentally-Derived **M**embranes of **A**rbitrary **S**hape Builder), which is designed to take experimental lipidomics and structural (e.g., electron microscopy and tomography) data as input and use them to build MD-ready models of cellular membrane systems. To test xMAS Builder's capabilities, we have used it to build two models (one  $\sim 12.0$  million atoms and the other  $\sim 11.6$  million atoms) of a test system with a representative complex lipid composition and geometry. The two models, which differed only in their lipid packing densities, both maintained their membrane integrity during an extended MD simulation (250 ns and

386 ns), but their highly divergent relaxation dynamics indicate that the proper packing density of curved membranes is determined by leaflet volume rather than surface area. These results suggest that xMAS Builder's algorithms produce high quality models and that simulation of these models will provide profound biophysical insights into the behavior of cellular membranes.

# 1 Introduction

Molecular dynamics (MD) is a powerful computational technique whose basic purpose is to elucidate the fundamental structural underpinnings of biological function through high spatial and temporal resolution simulations of the dynamics of biological systems. At present, MD has most commonly been used to study a single protein or protein complex at a time, but MD simulations have also successfully been applied to larger and more complex biological systems, including several viruses,<sup>1–10</sup> a chromatophore,<sup>11</sup> cellular cytoplasm,<sup>12</sup> a gene locus,<sup>13</sup> and even a minimal cell.<sup>14, 15</sup> As increasingly powerful high performance computing systems become available, there is a new opportunity to bring the unique capabilities of MD to new classes of cell-scale biological systems.

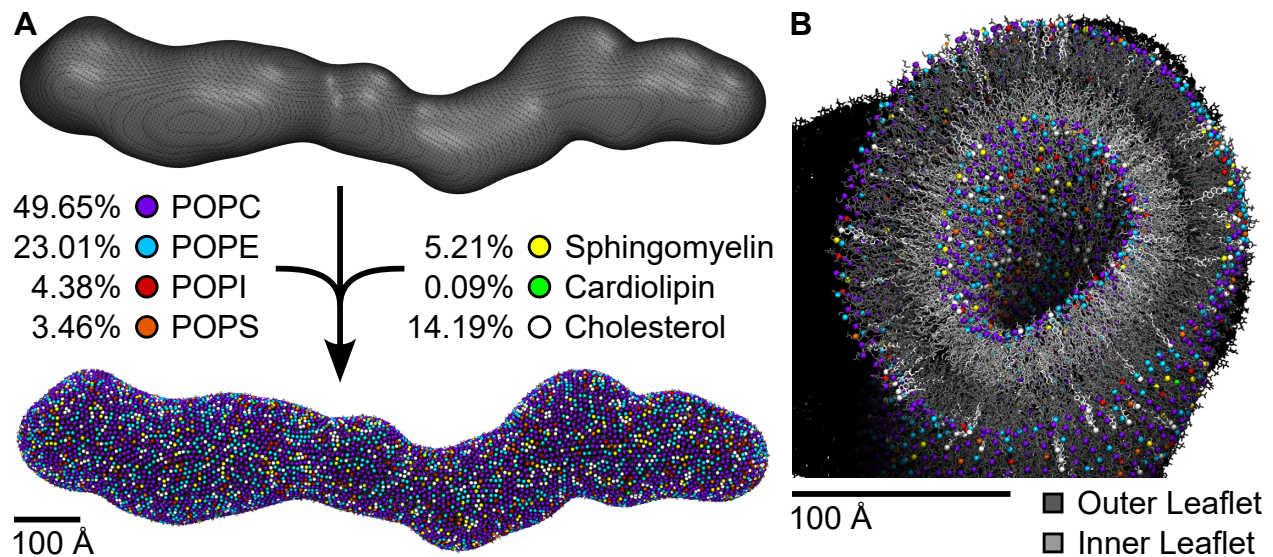
Cell-scale membranes are prime targets for detailed characterization with MD because they are a ubiquitous part of living systems, and they play an essential role in biological function, acting as semipermeable barriers that separate and protect cells and organelles from their surroundings. Within the cell, membranes adopt a variety of highly complex shapes, captured in detail by electron microscopy (EM) and electron tomography (ET) experiments,<sup>16–18</sup> and they are also highly complex in terms of their lipid and protein compositions. Combined with their immense scale, these features of cell-scale membranes pose a grand challenge to atomistic modeling, requiring a significant investment in the development of new modeling tools and techniques.

To meet this challenge, we have developed xMAS Builder (EXperimentally-Derived

**Membranes of Arbitrary Shape Builder**), a collection of scalable modeling algorithms designed to generate MD-ready atomistic models of cellular membrane structures. Starting with a triangle mesh derived directly from EM or ET data, xMAS Builder creates a membrane model by filling the arbitrarily complex mesh shape with an arbitrarily complex lipid composition derived from lipidomics data (Fig. 1). To ensure the final membrane model can be stably simulated with MD, xMAS Builder employs a series of modeling approaches to determine the proper thickness and packing density of the membrane, to evenly distribute the lipids over the surfaces of the membrane, to resolve complex steric clashes between lipids placed in the model, and to solvate and equilibrate the model.

xMAS Builder is not unique in its goal to generate MD-ready models of membrane systems. Many tools have been developed to generate membrane models with simple planar or spherical shapes, including CHARMM-GUI,<sup>19–24</sup> insane,<sup>25</sup> PACKMOL-Memgen,<sup>26</sup> and others. Other tools, namely PACKMOL<sup>27</sup> and BUMPY,<sup>28</sup> are designed to generate more complex membrane geometries through user-specified combinations of a handful of simple parametric shapes. cellPACK<sup>29</sup> is capable of packing lipids and proteins into mesoscale membrane models, but the significant challenge of adapting these models for use in MD simulations has not been addressed. Ultimately, only two tools, LipidWrapper<sup>30</sup> and TS2CG,<sup>31</sup> are capable of generating MD-ready membrane models with arbitrarily complex shapes, and we will discuss how they compare to xMAS Builder after describing its approach in detail.

xMAS Builder's modeling approaches have all been designed to handle cell-scale membrane models, but it was impractical to rapidly develop, troubleshoot, and test the efficacy of new approaches using a full cell-scale system. Instead, we used a smaller ( $\sim 1,124$  Å by  $\sim 260$  Å by  $\sim 268$  Å) synthetic test system with a realistically complex membrane shape, and we filled it using a lipid composition of the endoplasmic reticulum (ER) derived from experimental data<sup>32</sup> (Fig. 1). We ultimately produced two models of the test system, one containing  $\sim 12.0$  million atoms and the other  $\sim 11.6$  million atoms, that dif-



**Figure 1: Application of xMAS Builder to a test membrane system.** **(A)** xMAS Builder uses as input a mesh representing the surface of the membrane and a target lipid composition and turns them into an atomistic model that can be stably simulated using MD. In the application highlighted here, the input mesh is synthetic and is meant to capture the complex curvature of natural cellular membranes while being much smaller than them to make the development of xMAS Builder tractable. The input lipid composition is derived from an experimentally-determined lipid composition of the ER and is meant to be representative of the compositional complexity of cellular membranes. **(B)** Cutaway molecular rendering of the model generated by xMAS Builder viewed down its long axis. For each lipid, one atom near the membrane surfaces is represented as a sphere (the oxygen atom for cholesterol, the central linking carbon atom for cardiolipin, and the phosphorus atom for POPC, POPE, POPI, POPS, and sphingomyelin). In (A), the spheres have double the radius to better highlight the positions and identities of individual lipids.



ferred only in their lipid packing densities, and we evaluated the stability and quality of the models using long-term (250 ns and 386 ns, respectively) MD simulations. Here, we describe each of the new modeling approaches we developed for xMAS Builder, we report the results of applying them to the test system, and we discuss the results from the long-term MD simulations of the two versions of the model.

## 2 Results

xMAS Builder's approach to building models of cellular membranes has four major modeling stages:

1. Determine the lipid packing density and thickness of the membrane.
2. Place the lipids within the membrane.
3. Detect and eliminate complex steric clashes between lipids.
4. Solvate and equilibrate the membrane model.

For each of these modeling stages, we have developed new algorithms designed to tackle the unique challenges associated with building cellular membrane models, and, whenever appropriate, we have also applied existing algorithms in new ways. xMAS Builder is implemented as a series of Python 3 and VMD<sup>33</sup> TCL scripts. Several modeling steps involve running MD simulations with NAMD,<sup>34</sup> and we also make use of several standard triangle mesh algorithms implemented in MeshLab.<sup>35</sup>

In the rest of this section, we describe what each of xMAS Builder's modeling algorithms do and why they are designed and/or applied in the way that they are. We have used xMAS Builder to generate two models of a test system, and as we introduce each modeling algorithm, we also discuss its application to one of these models. Later, we introduce the other, earlier version of the model, discuss the application of xMAS Builder's modeling approaches to it, and compare it to the final model.

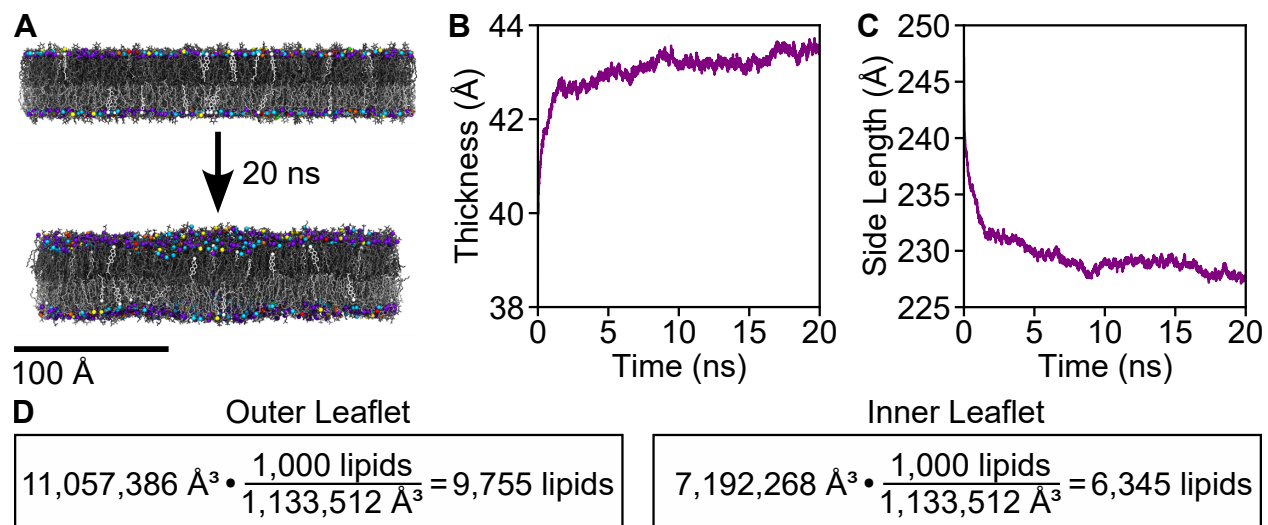
## 2.1 Determining the Lipid Packing Density and Membrane Thickness

When building a model with xMAS Builder, the user starts by providing two principal inputs: the desired lipid composition of the membrane model and a triangle mesh representing the shape of the membrane structure that is being modeled. This triangle mesh may be obtained from experimental sources (e.g., EM or ET) or generated synthetically, depending on the desired application. Together, these two inputs give the ultimate membrane model its defining characteristics, one of which is the number of lipid molecules present in the membrane model. Inaccurately estimating the number of lipids needed to fill a curved membrane with a complex lipid composition results in membrane defects when the model is simulated (see Sec. 2.6), so xMAS Builder's protocol is designed to ensure the accuracy of this estimate.

### 2.1.1 Approach

To start, an atomistic model of a planar membrane patch large enough to have a composition that is representative of the target lipid composition of the final model is prepared using CHARMM-GUI. It is then equilibrated with MD using the same simulation parameters that will be used in the final simulation of the membrane model. The point of this simulation is to determine the bulk properties of a membrane patch with the desired lipid composition, particularly its equilibrated volume and thickness.

Next, using the original mesh input and the measured thickness of the membrane patch, xMAS Builder generates meshes representing the inner and outer surfaces of the membrane as well as a surface that passes through the midplane of the membrane. This is done by sliding the original mesh vertices the appropriate distance along the vertex normal vectors while maintaining the original connectivity between the mesh vertices. A standard mesh algorithm (as implemented by the "Compute Geometric Measures" filter in MeshLab) is used to measure the volume enclosed by each of the generated meshes, and these volumes are in turn used to estimate the volume occupied by the inner and



**Figure 2: Determining the lipid packing density and thickness of the membrane. (A)** Molecular renderings of a planar membrane patch with the target lipid composition before and after 20 ns of simulation. The point of this simulation was to determine the physical parameters to use for the membrane model, which change depending on the target lipid composition. The same colors and molecular representations established in Fig. 1 are also used here. **(B)** Plot of the thickness of the membrane as a function of time during the simulation. This measurement determines how far apart lipids in the membrane's two leaflets are placed when the model is initially built. **(C)** Plot of the side length of the membrane patch as a function of time during the simulation. Together with the thickness measurement, this measurement determines the volume of the patch, which in turn determines the lipid packing density used for the membrane model. **(D)** Calculations of the number of lipids used to fill the inner and outer leaflets in the model, determined by multiplying the volume of the inner and outer leaflets of the membrane by the lipid packing density of the equilibrated membrane patch.

outer leaflets of the membrane model. Using these volumes, the measured volume of the membrane patch, and the input lipid composition, the number of lipids of each type needed to fill the inner and outer leaflets is estimated.

### 2.1.2 Application

The triangle mesh and lipid composition of the test membrane system being modeled here have been previously introduced (Fig. 1). Due to the small percentage of cardiolipin in the lipid composition, we generated a large membrane patch composed of 1,000 lipids per leaflet (Fig. 2A). We equilibrated the membrane patch for 20 ns, and the thickness and volume of the membrane stabilized after  $\sim 10$  ns to  $\sim 43.2$  Å and  $\sim 2,267,024$  Å<sup>3</sup>, respectively (Fig. 2B-C). After xMAS Builder generated the meshes representing the outer and inner surfaces and midplane of the membrane model, it estimated the volume occupied by the outer and inner leaflets to be  $\sim 11,057,386$  Å<sup>3</sup> and  $\sim 7,192,268$  Å<sup>3</sup>, respectively. From these numbers, xMAS Builder calculated that 9,755 and 6,345 lipids were needed to fill the outer and inner leaflets (Fig. 2D).

## 2.2 Optimizing the Initial Placement of the Lipids

Once the number of lipids needed to fill the inner and outer leaflets has been calculated, xMAS Builder must place the lipids within the leaflets. The most straightforward way to do this is to use the positions of vertices from the meshes representing the inner and outer surfaces of the membrane. This approach requires that the meshes have at least as many vertices as there are lipids in the model, which can be achieved using standard mesh upsampling algorithms if needed. Unfortunately, this direct approach also introduces some consequential artifacts into the membrane model.

One problem is that mesh vertices are not uniformly distributed over the surface of a mesh, so using this approach results in an uneven initial packing of the placed lipids. Additionally, it is difficult to control the exact number of vertices within a mesh, so meshes with

slightly more vertices than there are lipids have to be used in practice. As such, not every vertex has a lipid placed at its position, creating small voids that compound the uneven distribution of the placed lipids. Given enough equilibration time, these artifacts would dissipate. However, because of the large scale of the models xMAS Builder is designed to build, it is essential to minimize the amount of equilibration that must be performed on the full membrane model before it can start producing meaningful results. Accordingly, we have designed a new scalable and computationally efficient approach to optimize the initial placement of the lipids with xMAS Builder.

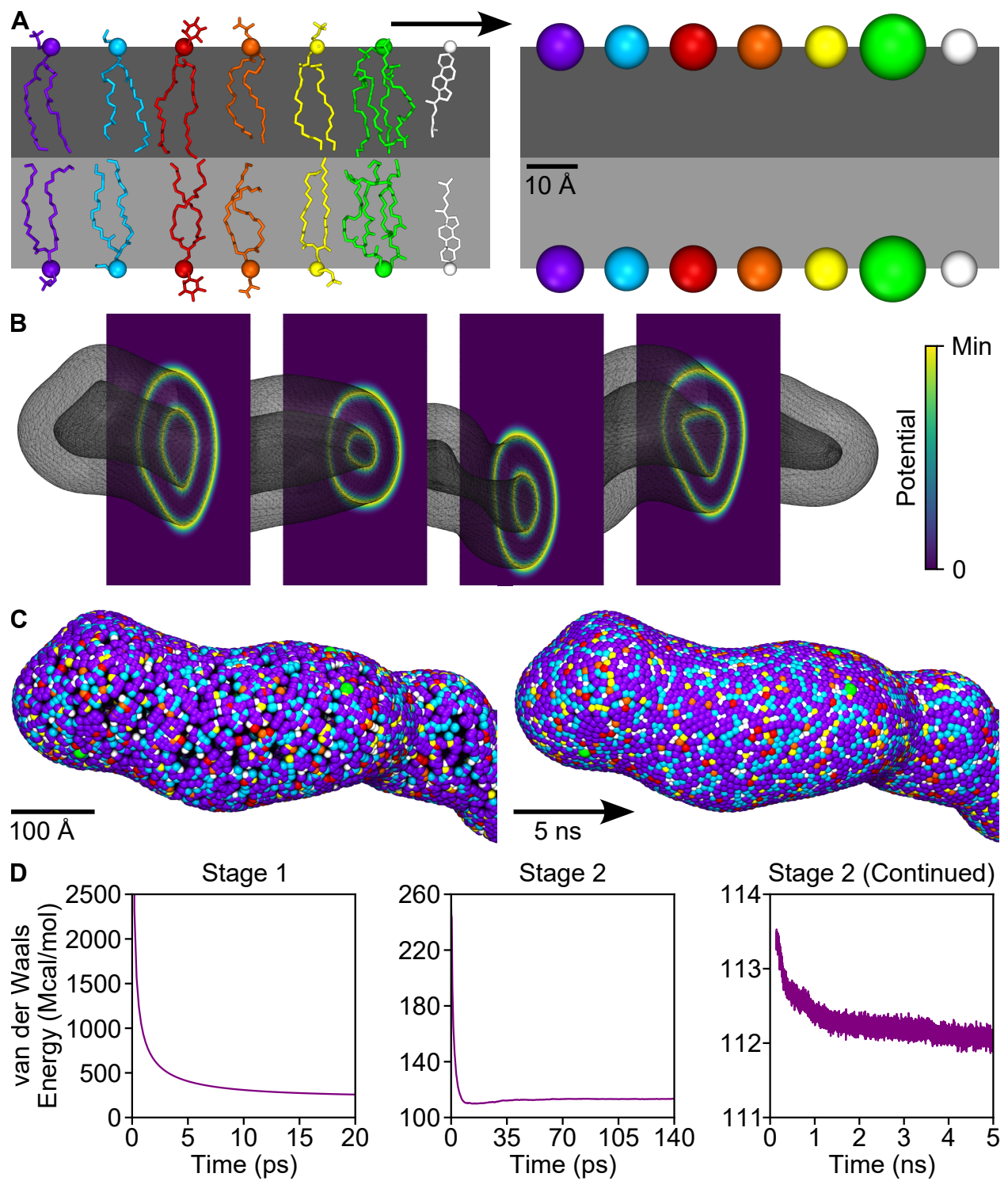
## 2.2.1 Approach

In xMAS Builder's approach, a single particle is used to represent each lipid in the system (Fig. 3A). The particles are initially placed on the vertices of the meshes representing the inner and outer surfaces of the membrane, meaning they are not initially evenly distributed on the surfaces of the meshes. To optimize their positions, they are simulated using MD while restrained to the inner and outer surfaces of the membrane model. Basing xMAS Builder's approach on an MD simulation allows it to harness the established parallel performance of NAMD to optimize the initial positions of millions of lipids at once, and the small particle count of the simulation gives it a highly economical computational cost.

In the simulation, the particles are designed to interact with one another solely through Lennard-Jones potentials (as specified by the CHARMM36m force field<sup>36–39</sup>), and different parameters ( $\epsilon_i$  and  $r_i^{\min}$  below) are used for each type of lipid in the model.

$$U_{ij}^{LJ} = \epsilon_{ij} \left( \left( \frac{r_{ij}^{\min}}{r_{ij}} \right)^{12} - 2 \left( \frac{r_{ij}^{\min}}{r_{ij}} \right)^6 \right), \quad \epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}, \quad r_{ij}^{\min} = \frac{1}{2} (r_i^{\min} + r_j^{\min})$$

The chosen parameters are designed to ensure that the particles constantly experience van der Waals repulsion and to ensure that the space occupied by the particles is propor-





**Figure 3: Optimizing the initial placement of the lipids.** **(A)** The positions of the lipids is optimized using an MD simulation in which each lipid is represented as a single uncharged particle. The mass of the particle is determined by the mass of the lipid, and its radius is determined by the area-per-lipid value reported by CHARMM-GUI. The same colors and molecular representations established in Fig. 1 are also used here. **(B)** Visualization of the potential used by the Grid Forces feature of NAMD to restrain the lipid particles to the surfaces of the membrane during the simulation. The force applied by Grid Forces is proportional to the negative gradient of the user-specified potential, and the potential in this application has the functional form of a truncated Gaussian centered on the meshes representing the membrane surfaces. **(C)** Distribution of the lipid particles before and after the simulation. The initial positions of the lipid particles are determined by the positions of a random selection of vertices from the meshes representing the surfaces of the membrane. Large gaps can be seen between the particles at the start of the simulation, but the particles evenly distribute themselves over the membrane surface over the course of the simulation. **(D)** Plots of the total particle interaction energy during the simulation, which is used to determine when the distribution of the particles has stabilized. The simulation occurs in two stages. The first stage (20 ps) uses the velocity quenching feature of NAMD to eliminate large overlaps between particles while preventing them from moving too quickly and destabilizing the simulation. The second stage (5 ns) proceeds using standard MD.

tional to the space occupied by the lipids they represent. As such, all of the particles are given the same value of  $\epsilon_i$  (empirically set at  $-0.3 \frac{\text{kcal}}{\text{Mol}}$ ), and  $r_i^{\text{min}}$  is set for each particle such that it is twice the radius of a circle with an area equal to the pertinent area-per-lipid value reported by CHARMM-GUI.

The particles are restrained to the inner and outer surfaces of the membrane using the Grid Forces<sup>40</sup> feature of NAMD. In Grid Forces, an energetic potential field is created over a region of a model by assigning potential values to points on a regular three dimensional grid. Between grid points, the potential is determined using an interpolation scheme, and the force applied to each particle is determined by calculating the negative gradient of the potential multiplied by a scaling factor. To use Grid Forces, xMAS Builder generates potential grids that represent the shapes of the inner and outer surfaces of the membrane (Fig. 3B). Specifically, xMAS Builder does this by creating an empty grid over each mesh, calculating the minimum distances between each grid point and the mesh, applying a truncated Gaussian function to each of those distances, and assigning the resulting values to the original respective grid points.

With the particles representing the lipids placed and Grid Forces active, the MD simulation can begin. Due to the strength of the interaction forces between the particles, the poor initial placement of the particles makes standard MD unstable for the initial configuration of the system. As such, xMAS Builder runs the simulation at a temperature of 100 K, and standard MD is preceded by a short MD simulation with the velocity quenching feature of NAMD active, which simply prevents particles from moving more than a set distance within a time step. MD with velocity quenching is run until the interaction energy between the particles stabilizes, and standard MD is then also run until the interaction energy between the particles re-stabilizes.

After both stages of the simulation is complete, the particles must be replaced with atomistic lipids. To determine the orientations to use for the atomistic lipids, the simulation particles are separated into two groups and treated as two point clouds (one for the inner

surface and one for the outer surface), and a standard mesh algorithm (as implemented by the “Compute normals for point sets” filter in MeshLab) is applied. The particles are then replaced with atomistic lipids from a library of lipid conformations derived from CHARMM-GUI, which produces a fully atomistic model of the membrane system.

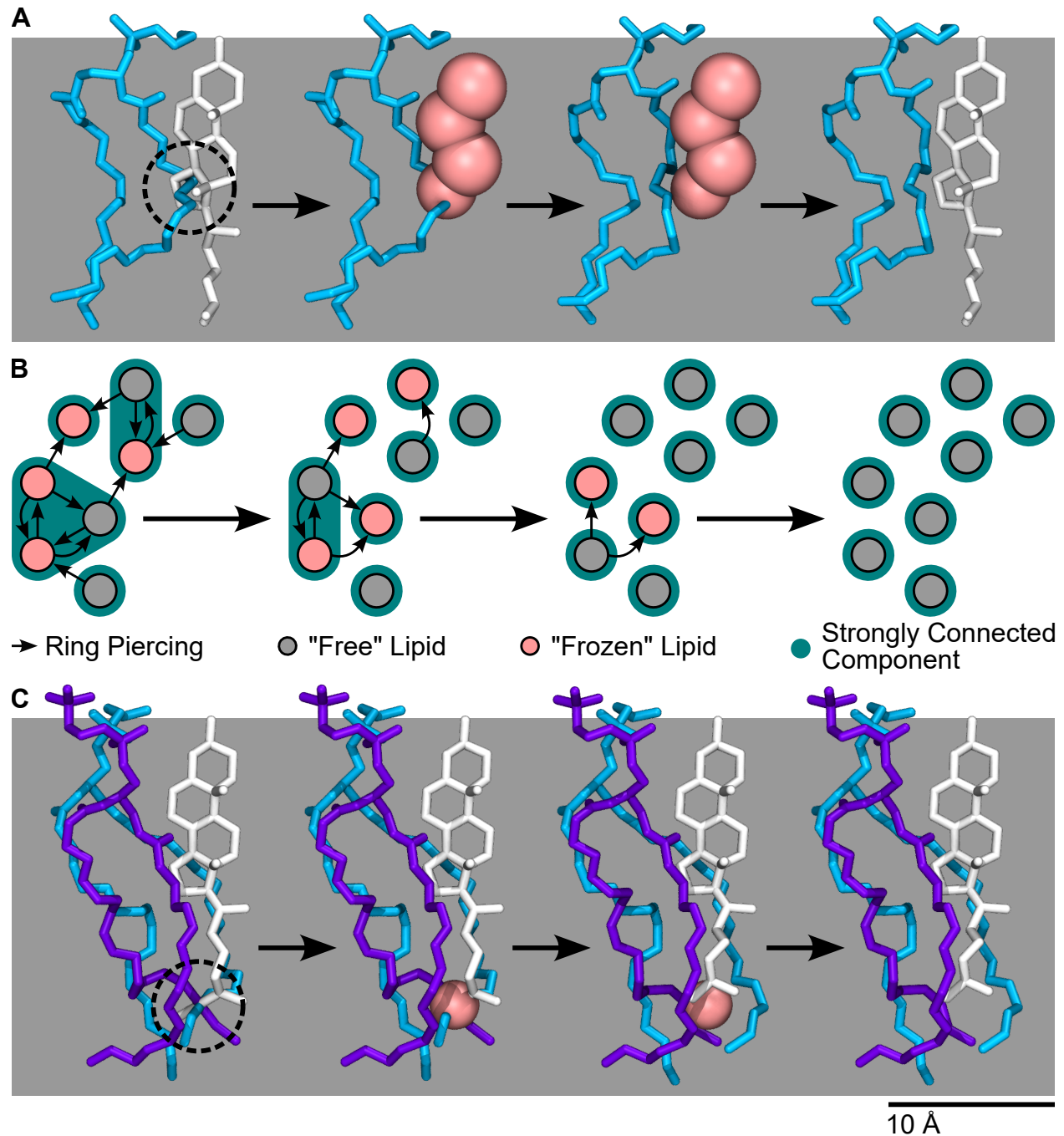
## 2.2.2 Application

After xMAS Builder generated the Grid Forces grids and placed the particles representing the lipids (Fig. 3B-C), MD with velocity quenching was performed for 20 ps, causing the interaction energy between the particles to drop from an initial value of  $\sim 2,326 \frac{\text{Mcal}}{\text{Mol}}$  to a final value of  $\sim 257 \frac{\text{Mcal}}{\text{Mol}}$  (Fig. 3D). Following MD with velocity quenching, standard MD was performed for 5 ns, during which time the interaction energy dropped to a stable value of  $\sim 112 \frac{\text{Mcal}}{\text{Mol}}$ . As intended, this simulation produced an even distribution of the lipid particles over the surfaces of the membrane model (Fig. 3C).

## 2.3 Fixing Complex Steric Clashes Among the Lipids

After the simulation to optimize the initial placement of the lipids, xMAS Builder uses the positions of the particles to generate a fully atomistic model of the membrane system. However, this model is not yet suitable for MD simulation because of the existence of complex steric clashes between the lipids that make up the model, namely ring piercings and lipid entanglements. A ring piercing occurs when a bond from one lipid passes through the center of a ring in another lipid, and a lipid entanglement occurs when three sets of bonded atoms are locked together in the same region of space (Fig. 4). xMAS Builder must fix these clashes before an MD simulation can be run because they do not occur in natural membrane systems, they cause MD simulations to crash, and they cannot be eliminated by applying standard MD minimization.

Complex lipid clashes are introduced into the membrane model because xMAS Builder places individual lipids blindly (i.e., without regard for the placement of other lipids). It



**Figure 4: Fixing complex steric clashes between lipids. (A)** Strategy to fix ring piercings. In the initial configuration, a POPE lipid (cyan) pierces the ring of a cholesterol lipid (white). To fix this piercing, the cholesterol lipid is removed from the membrane, and a particle (pink) representing each of its rings is added to the membrane. Then, 500 steps of minimization are performed in NAMD while keeping the positions of the ring particles fixed, which causes the POPE tail to no longer overlap with the particles. The particles are then removed from the membrane, the cholesterol lipid is added back with its original conformation, and the new conformation of the POPE lipid no longer pierces the ring of cholesterol. Because their conformation does not change during this process, we refer to the ringed lipids replaced by particles as “frozen,” and we refer to the rest of the lipids as “free.” **(B)** Strategy to handle networks of ring piercings. xMAS Builder represents networks of ring piercings as directed graphs, with nodes representing lipids and with edges directed from the piercing lipid to the pierced lipid. Within the directed graphs, xMAS Builder identifies the strongly connected components and “freezes” all but one of the lipids within the component. The lipid chosen to be “free” is the one that is pierced by the greatest number of lipids and that is not also pierced by any lipid from outside of the strongly connected component. With the “frozen” and “free” lipids selected, xMAS Builder applies the approach from (A), and networks of ring piercings are searched for again in the resulting system. These steps are applied iteratively until no ring piercings are detected in the system. **(C)** Strategy to correct lipid entanglements. In the initial configuration, bonds from a cholesterol lipid (white), a POPC lipid (purple), and a POPE lipid (cyan) are unnaturally extended and centered on the same region in space, creating a lipid entanglement. To correct this issue, xMAS Builder places a particle (pink) at the center of the entanglement and holds its position fixed during 500 steps of minimization in NAMD. The presence of the particle disrupts the energetic favorability of the entangled conformation, allowing the lipids’ atoms to move away from each other and the bonds to return to their natural lengths during minimization. At the end of this process, the particle is removed.

is possible, of course, to place lipids using a non-blind approach (e.g., as in CHARMM-GUI's<sup>19–24</sup> approach), but such approaches do not scale to cell-sized membrane systems. Moreover, existing approaches to fix ring piercings sometimes fail, forcing the user to fix the remaining problems manually or rebuild the entire system from scratch. Occasional failure is fine for smaller membrane models, but it is not a viable option for cell-scale systems where the number of ring piercings is orders of magnitude larger.

Standard MD minimization is able to fix simple atomic clashes present in newly built models, and it is designed to efficiently harness the parallel computing power of MD engines. However, it is not able to correct ring piercings and lipid entanglements because, in both of these molecular configurations, opposing atomic interactions counterbalance each other. Specifically, steric clashes between the molecular groups in the system work to push the groups apart, while covalent bonds between some groups work to pull them back together. As such, to disrupt these configurations, the interaction energies between the groups need to temporarily increase through worsened steric clashes and/or lengthened covalent bonds, which minimization cannot achieve on its own because it is designed to exclusively lower interaction energies. In xMAS Builder, we have therefore developed a customized approach to fix these complex lipid clashes.

### 2.3.1 Approach

Before ring piercings can be corrected, they must first be detected. In xMAS Builder's approach, rings in lipids are decomposed into triangles, nearby bonds from other lipids are treated as line segments, and simple geometric tests (modified from<sup>41</sup>) are applied to determine if any of the line segments intersect any of the triangles. After ring piercings have been detected, they are corrected by removing lipids with pierced rings from the system, replacing each pierced ring with a large particle, holding the position of the particle fixed, and applying 500 steps of minimization to the full system (Fig. 4A). The presence of the particle representing the ring makes it energetically unfavorable for atoms to intersect



with the particle, allowing minimization to gently push them away from the ring. After minimization, the particles are replaced with the original atomistic lipids. Because the conformations of the lipids replaced with particles do not change throughout this process while the conformations of the other lipids do, we refer to the replaced lipids as “frozen,” and the rest of the lipids as “free.”

This process is complicated by the existence of networks of ring piercings (i.e., collections of ringed lipids piercing each other in a cyclical manner). The simplest example of a network of ring piercings involves two lipids with two interlocked rings. If the rings of these two lipids were both replaced with particles, the subsequent minimization would accomplish nothing because the position of both particles is held fixed, and none of the atoms involved in the original piercings would be present during minimization. To handle this situation, xMAS Builder represents all ring piercings detected within a system as a directed graph, finds the connected components within the graphs (which represent the networks), finds the strongly connected components within the graph’s connected components, “freezes” all but one lipid within each strongly connected component, and applies minimization (Fig. 4B). This approach breaks networks up into multiple pieces, which are often themselves smaller networks. As such, the procedure to detect and break up networks must be applied iteratively until xMAS Builder no longer detects any ring piercings in the system.

As with ring piercings, lipid entanglements must first be detected before they can be corrected. When minimization is applied to the whole model to correct ring piercings, the bonds involved in an entanglement become highly extended, and they center themselves on the same region in space (Fig. 4C), making them straightforward to detect. Once found, xMAS Builder places a large particle at the center of each entanglement, and the position of the particle is held fixed during 500 steps of minimization. The presence of the particle creates a steric clash between it and the atoms involved in the entanglement, destabilizing the entanglement and allowing minimization to move the atoms involved away from the

location of the fixed particle. After applying minimization, the large particles are removed from the system.

### **2.3.2 Application**

The test system is composed of 16,100 lipids, and 2,990 of those lipids contain rings (i.e., cholesterol and POPI). xMAS Builder took two iterations of 500 steps of minimization to resolve all of the ring piercings in the test system. Specifically, it detected 805 ring piercings in the initial system, then 12 after the first minimization iteration, then 0 after the second. For the first iteration, it selected 437 lipids to “freeze,” then it selected 7 to “freeze” for the second iteration. After correcting the ring piercings, xMAS Builder detected that there were 3 lipid entanglements in the system, which it then resolved in 500 steps of minimization. Thus, xMAS Builder corrected all of the complex lipid clashes in the system with a total of 1,500 steps of minimization, a tiny fraction of the millions of steps of MD used in the ultimate simulation of the system.

## **2.4 Solvating and Equilibrating the Membrane Model**

After the complex steric clashes among the lipids have been resolved, the membrane is ready to be solvated, which presents a unique challenge to xMAS Builder. In a typical MD simulation, water simply surrounds the system of interest, and inaccuracies in the number of water molecules placed are easily accommodated by the variable simulation box size used in a constant-pressure MD simulation. In contrast, water both surrounds and is enclosed by the membranes modeled by xMAS Builder, and inaccuracies in the number of water molecules placed in the enclosed volume result in the membrane collapsing in on itself or bursting.

Because water occupies the spaces between the head groups of lipids, the number of molecules needed to solvate a membrane depends on both the composition and curvature of that membrane. The exact nature of this dependence is the subject of active

investigation, but for now, we do not have a way to precisely estimate the proper number of water molecules needed to solvate a curved membrane. Thus, by necessity, xMAS Builder's approach to solvating its membrane models involves a restrained equilibration MD simulation in which inaccuracies in the initial water molecule count reveal themselves, are quantified, and are corrected.

Running an MD simulation of the full-size model just to correct inaccuracies in the initial water molecule count would be computationally very costly, but it is fortunately possible to also equilibrate the membrane while running this simulation. Simulating the membrane while its overall shape is restrained allows the membrane's lipids to adopt a natural packing both against one another and within the target shape of the model, and it is essential to allow this process to happen to ensure a stable final model that produces meaningful results. To set the length of the restrained equilibration simulation, we estimate that the amount of time needed to equilibrate the lipids within their local environment is approximately the same as the amount of time needed to equilibrate a membrane patch with the same target lipid composition.

### **2.4.1 Approach**

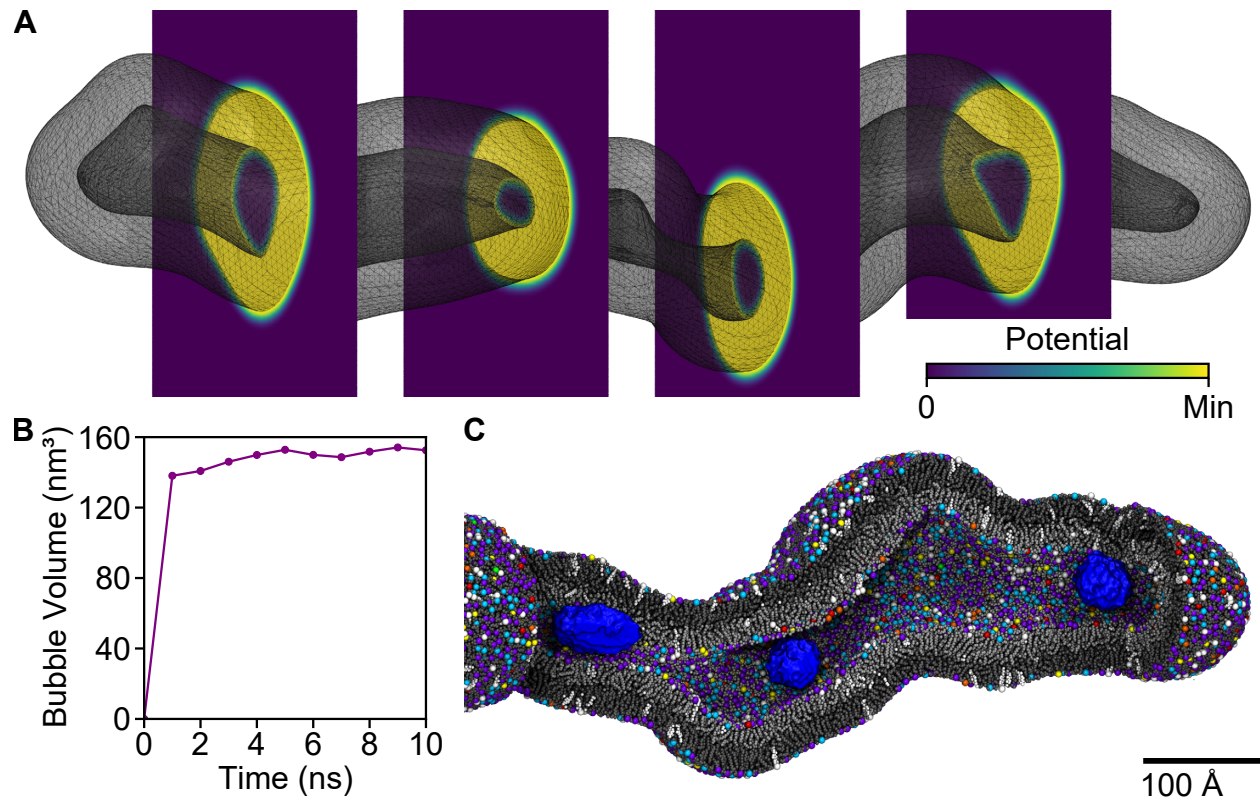
To solvate a system, xMAS Builder starts by designating a simulation box size that encloses the entire membrane, and it measures the volume enclosed by the inner surface of the membrane as well as the volume between the edges of the box and the outer surface of the membrane. Using the known density of water, xMAS Builder places the theoretically appropriate number of water molecules in both volumes, and there is then an option to add ions. Independently for the internal and external water volumes, the user specifies which ions are desired along with their concentrations, and the user specifies whether ions should be used to neutralize the membrane. xMAS Builder then calculates how many ions of each species need to be placed in each volume, and a corresponding selection of water molecules are replaced with the appropriate ions.

With the water and ions in place, xMAS Builder next sets up a restrained equilibration simulation, which gives the lipids and water a chance to adjust to the desired shape of the membrane. Using the Grid Forces feature of NAMD (introduced previously), the overall shape of the membrane is maintained, and water is prevented from passing across the membrane. Near the boundaries of the membrane, the functional form of the potential is a truncated half Gaussian, identical to what was used in the simulation to optimize the initial placement of the lipids. However, between the boundaries of the membrane, the potential is flat, meaning no forces are exerted on the core of the membrane (Fig. 5A). The form of the restraint placed on the water is similar, but the potential is flat in the bulk of the solvent, and a truncated half Gaussian acts on the solvent within the head group region of the membrane.

As the membrane equilibrates, water molecules fill in the spaces between the head groups of the lipids, making xMAS Builders' initial estimates of the number of molecules needed for solvation consistently low. Restrained equilibration of the system thus consistently results in vacuum bubbles forming in the water, and the volume of these bubbles is monitored as the simulation progresses. After their volume has stabilized, the bubbles are filled with water and ions, and the appropriate number of water molecules needed for this purpose can now be accurately estimated since the membrane has been fully solvated.

## 2.4.2 Application

Using xMAS Builder, we solvated the membrane of the test system, we added 0.15 M NaCl to the water, and we performed 10 ns of restrained equilibration. The length of the simulation was determined by the  $\sim 10$  ns of simulation needed to equilibrate the planar membrane patch (as indicated by the stabilization of its thickness and side length, Fig. 2). During this time, three vacuum bubbles formed in the interior of the system, and their volume stabilized after  $\sim 5$  ns (Fig. 5). We used xMAS Builder to fill these vacuum bubbles with additional water and ions, bringing the final system size to  $\sim 11.6$  million atoms, and we



**Figure 5: Solvating the membrane model.** (A) Visualization of the potential used by the Grid Forces feature of NAMD to restrain the shape of the membrane during equilibration. Just outside the membrane, the functional form of the potential is a truncated half Gaussian centered on the meshes representing the membrane surfaces. Within the membrane, the potential has a uniform value. The force applied by Grid Forces is proportional to the negative gradient of the potential, so the membrane experiences no restraining force except when it pushes against the boundaries defining its target shape. (B) Plot of the volume of the vacuum bubbles that form in the water as a function of time. As water packs around lipid head groups in the membrane, not enough water is left to fill the volume enclosed by the membrane, and vacuum bubbles form. Once the volume of the vacuum bubbles has stabilized, xMAS Builder fills them with additional water and ions. (C) Cutaway molecular rendering of the vacuum bubbles (blue) that form inside the membrane model after 10 ns of equilibration. For the membrane, the same colors and molecular representations established in Fig. 1 are also used here.

then allowed the solvent to equilibrate during an additional 1 ns of restrained equilibration.

## 2.5 Simulating the Membrane Model

Following the restrained equilibration simulation, we released all restraints on the membrane and allowed it to equilibrate under unbiased conditions for 386 ns (Fig. 6). Over the course of the unrestrained simulation, the overall shape of the membrane relaxed, with areas of high local curvature becoming smoother over time. This result was expected as there was no asymmetric membrane composition or proteins present to maintain the curvature of the system. By the end of the simulation, the membrane had adopted mostly flat or semi-circular cross-sections, which makes sense as these shapes alleviate the tension caused by the unequal number of lipids in the two leaflets. Because the membrane's global shape changed significantly over time, the simulation also tested the strength and resilience of the membrane's construction, acting as a stress-test for xMAS Builder's modeling protocols. While its shape relaxed, the membrane's integrity was never compromised by any pores or fissures, indicating that the lipids were evenly distributed within the membrane and that the proper amount of water filled the model's interior.

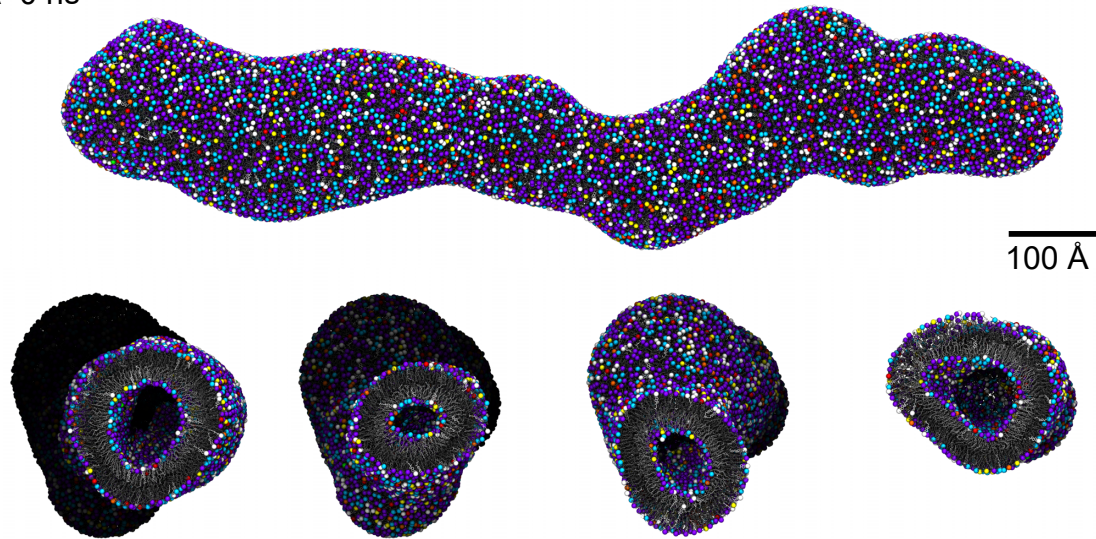
## 2.6 Building and Simulating an Earlier Version of the Membrane Model

Prior to the version of the membrane model that has been discussed up to now, we used xMAS Builder to produce an earlier version of the model. The earlier model used the same shape and lipid composition and was built using the same modeling steps, but xMAS Builder used calculations based on surface area rather than volume to determine the lipid packing density (Fig. S1). This approach resulted in a model that contained 1,797 more lipids in the outer leaflet and 1,143 fewer lipids in the inner leaflet compared to the final volume-based model.

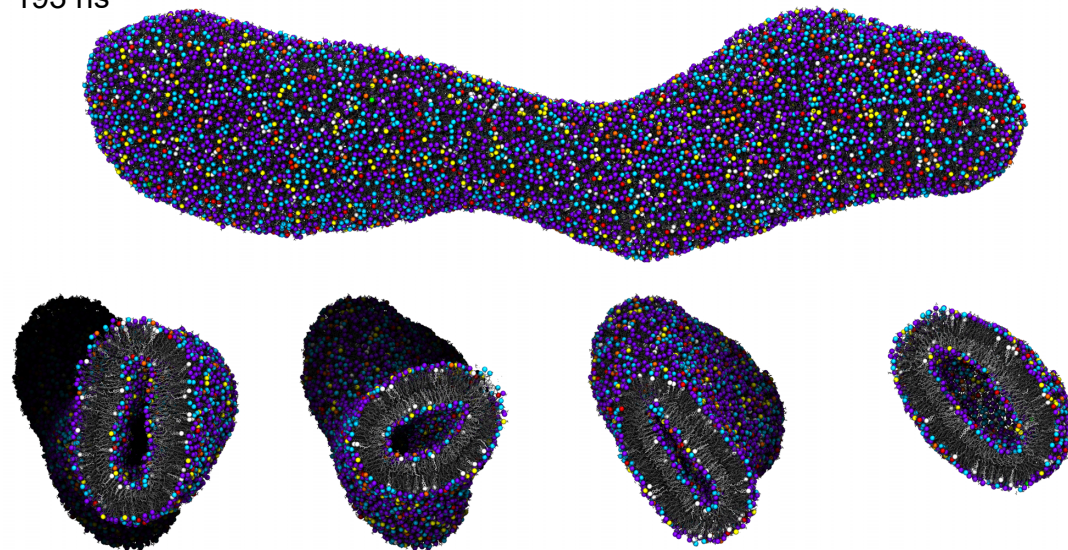
When simulated under unbiased conditions for 250 ns, the membrane's integrity was



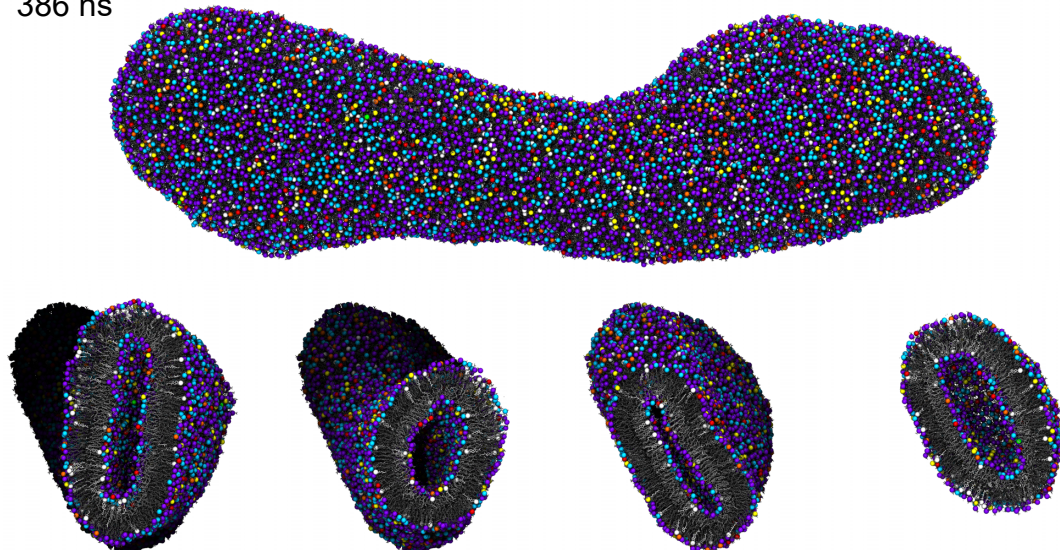
**A** 0 ns



**B** 193 ns



**C** 386 ns



**Figure 6: Simulating the membrane model. (A)** Molecular renderings of the membrane after restrained equilibration. The top panel shows the whole model, and the bottom panels show cutaway renderings of the model at regular intervals along its long axis. The same colors and molecular representations established in Fig. 1 are also used here. **(B)** Molecular renderings of the membrane halfway through the simulation (after 193 ns). With no proteins or asymmetric lipid composition to enforce a specific curvature, the shape of the membrane relaxed. The fine curvature features of the initial model eroded, and the membrane flattened as much as it could. **(C)** Molecular renderings of the membrane at the end of the simulation (after 386 ns). As its shape continued to relax, the membrane lost even more of its initial complex curvature. Importantly, throughout the entire simulation, the integrity of the membrane was never compromised by pores or fissures.

never compromised by the formation of any pores or fissures, again illustrating the robustness of xMAS Builder's modeling algorithms. The shape of the membrane also relaxed much like the volume-based model, with the membrane's initial fine curvature features dissipating over time (Fig. 7). Tension caused by the unequal number of lipids in the inner and outer leaflets was eased by the model adopting mostly circular cross-sections by the end of the simulation. However, this tension was also relieved by the formation of three bilayer protrusions from the outer leaflet, which grew in size as the simulation progressed.

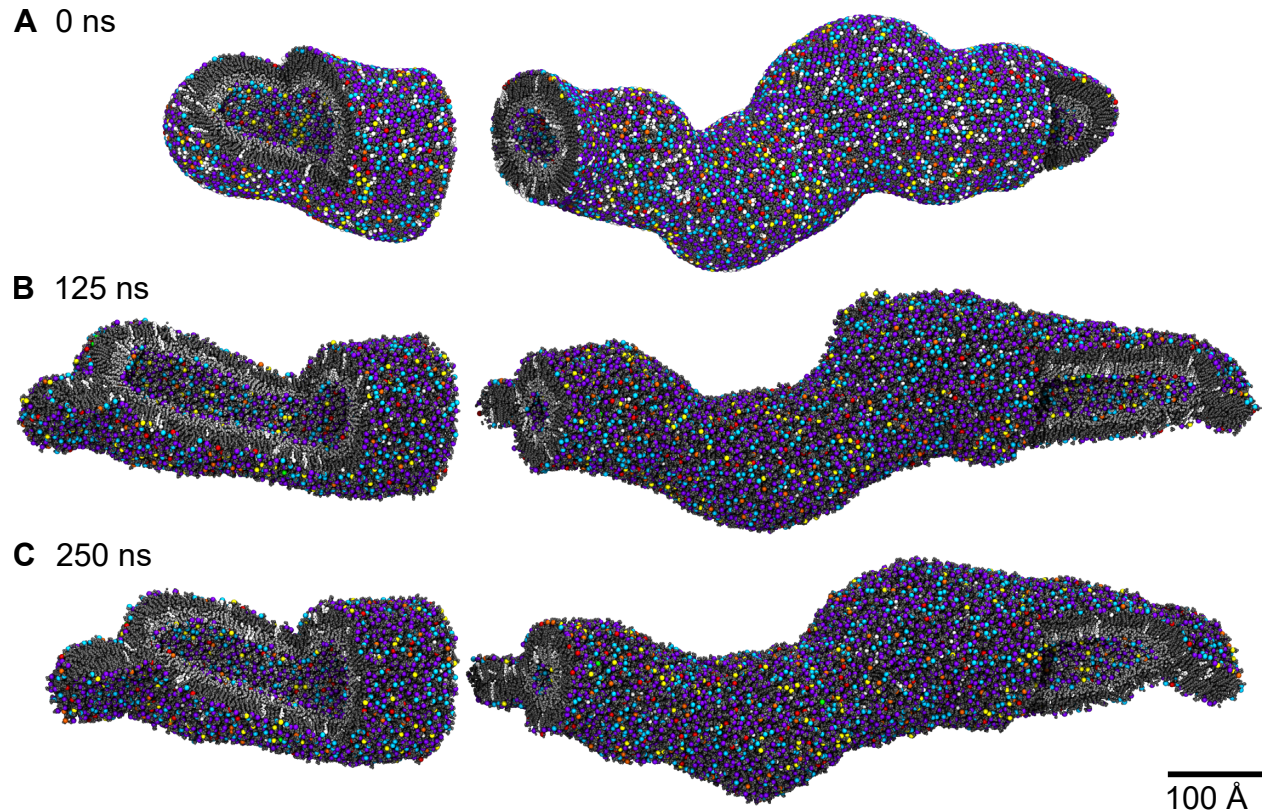
The formation of these protrusions strongly suggested to us that the outer leaflet of this model was significantly overfilled relative to the inner leaflet, and it prompted us to reevaluate our strategy to determine lipid packing density. The issue with basing packing density on surface area is that it only optimizes the packing density for the lipid head groups. In a curved membrane, this strategy leaves the lipid tails significantly over-packed or under-packed depending on the direction of the curvature, which explains why the outer leaflet was so overfilled relative to the inner leaflet in the largely tubular shape of the test system.

To address this issue, we adopted a new strategy for calculating lipid packing density based on volume, and we built a new model of the test system using it. Our intention was that this strategy would take into account the three dimensional nature of lipids and membranes, balancing slight over-packing at one end of a lipid with slight under-packing at the other. After extended equilibration of the volume-based model, there were no bilayer protrusions or any other indications of overfilled or underfilled leaflets, suggesting that this strategy accurately estimates the number of lipids needed to fill a curved membrane.

### 3 Discussion

In this work, we have developed xMAS Builder, a tool designed to build realistic atomistic models of cellular membranes suitable for MD simulations. xMAS Builder has been





**Figure 7: Simulating a previous version of the membrane model with a lipid packing density based on surface area. (A)** Cutaway molecular rendering of the membrane after restrained equilibration. This version of the membrane model was built to have exactly the same shape and size as the final model, but it had 1,797 more lipids in the outer leaflet and 1,143 fewer lipids in the inner leaflet because its target packing density was determined by surface area rather than volume. The same colors and molecular representations established in Fig. 1 are also used here. **(B)** Cutaway molecular rendering of the membrane halfway through the simulation (after 125 ns). As with the volume-based model, the shape of the membrane relaxed over the course of the simulation, eroding the fine curvature features of the initial shape. Additionally, in response to overcrowding, three bilayer protrusions from the outer leaflet formed (highlighted by the three cutaways in the rendering). **(C)** Cutaway molecular rendering of the membrane at the end of the simulation (after 250 ns). The shape of the membrane continued to relax, losing even more detail from its original shape, and the protrusions from the outer leaflet also continued to grow.

designed specifically to handle arbitrarily complex lipid compositions and geometries derived from experimental lipidomics and structural (e.g., EM or ET) data. The new modeling algorithms have also been designed to scale to cellular sizes, often harnessing the existing highly developed parallel computing power in NAMD to accomplish this goal. At the same time, these new modeling algorithms (e.g., the algorithm to fix ring piercings) are also useful for traditionally sized systems.

To develop and validate the modeling algorithms of xMAS Builder, we have applied it to a test system with a representative complex geometry and lipid composition. We built two models of the test system, one containing  $\sim 12.0$  M atoms with a lipid packing density determined by surface area and one containing  $\sim 11.6$  M atoms with a lipid packing density determined by volume, and we equilibrated them with 250 ns and 386 ns, respectively, of unbiased MD. The integrity of both membranes was maintained over time while their overall shapes relaxed, but three protrusions developed on the outer leaflet of the area-based model, indicating it was overfilled relative to the inner leaflet and suggesting that volume rather than surface area should be used to determine lipid packing density. These results demonstrate that the modeling techniques developed for xMAS Builder produce high quality models, and they illustrate that these models may be used to probe the fundamental biophysics governing the structure and function of cell-scale membrane systems.

Besides xMAS Builder, TS2CG<sup>31</sup> and LipidWrapper<sup>30</sup> are two other tools that have also been designed to build MD-ready models of membranes with arbitrarily complex shapes. While xMAS Builder has been designed to generate membrane models for atomistic MD, TS2CG's models are built for coarse-grained MD. The trade-offs between atomistic and coarse-grained MD simulations are well established (see,<sup>42</sup> for example), and we view xMAS Builder and TS2CG as complementary tools depending on what level of resolution is desired for a particular simulation. Of course, methods are also available to change the resolution of a model from atomistic to coarse-grained and vice versa (e.g., `martinize.py`<sup>43</sup> and `backward.py`<sup>44</sup> from the developers of the Martini force field<sup>45</sup>), but

we expect the current implementations of these methods will need to be adapted before they can handle cell-scale models.

Like xMAS Builder, LipidWrapper has been designed to generate models of membranes with arbitrarily complex shapes for atomistic MD, but the modeling approaches of the two tools are entirely different. In LipidWrapper's approach, equilibrated triangular membrane patches are used to fill the input mesh's triangles, and overlapping lipids at the triangle boundaries are simply removed from the model. This approach makes it difficult to control the lipid composition and packing density of the model, and extensive equilibration is required to heal the boundaries between the triangles and to allow the flat triangular patches to curve.<sup>7</sup> LipidWrapper also has no modeling protocols to handle the unique challenge of solvating closed membrane systems. Our results indicate that proper lipid packing density and solvation have a pronounced impact on the integrity and quality of membrane models with complex closed shapes, and we suggest that xMAS Builder has an advantage over LipidWrapper in these areas.

The present implementations of the modeling algorithms used by xMAS Builder are open source and will be released upon publication of this manuscript. In future work, we plan to improve the user friendliness of these implementations with the ultimate goal of producing an easy-to-use, self-contained software package that encapsulates all of the capabilities of xMAS Builder. We also plan to continue to develop these implementations to improve their efficiency and reduce bottlenecks in xMAS Builder's modeling protocols. Beyond improvements to the implementations of the existing modeling algorithms of xMAS Builder, we also plan to incorporate new modeling algorithms, already under development, which will allow xMAS Builder to include membrane proteins and soluble proteins in the models it builds. This additional capability will allow xMAS Builder to generate even more realistic and complex membrane models, enabling detailed characterization of the structural dynamics of these systems with MD simulations.



## 4 Methods

Unless otherwise specified, all MD simulations were performed with NAMD<sup>34</sup> using the CHARMM36m force field<sup>36–39</sup> at a constant temperature of 310 K and a constant pressure of 1 atm. Constant temperature was maintained using Langevin dynamics for all non-hydrogen atoms with a damping coefficient of 1 ps<sup>−1</sup>. Constant pressure was maintained using a Nosé-Hoover Langevin piston<sup>46,47</sup> with a period of 100 fs and a damping time scale of 50 fs. Simulations were performed with an integration time step of 2 fs. The cut-off distance for both electrostatic and van der Waals interactions was set to 12 Å, and a switching function was applied at 10 Å. Periodic boundary conditions were applied in all simulations, and long-range electrostatic interactions were calculated using the particle mesh Ewald method<sup>48,49</sup> with a grid point density of 1 Å<sup>−1</sup>.

## 5 Data Availability

All data generated or analyzed in this study are included in this manuscript and its supplementary information files.

## 6 Code Availability

The source code for xMAS Builder will be released upon publication of this manuscript.

## 7 Acknowledgments

Research reported in this publication was supported by grants from the National Institutes of Health under award numbers P41-GM104601 and R24-GM145965. NT acknowledges support from the National Science Foundation Graduate Research Fellowship Program under Grant No. 1746047. Any opinions, findings, and conclusions or

recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the National Institutes of Health or the National Science Foundation. This work used resources of the Extreme Science and Engineering Discovery Environment (XSEDE), which was supported by National Science Foundation grant number ACI-1548562, through allocations TG-MCA06N06 and TG-MCA93S028. This work also used resources of the Oak Ridge Leadership Computing Facility, which is a Department of Energy Office of Science User Facility supported under Contract DE-AC05-00OR22725, through Innovative and Novel Computational Impact on Theory and Experiment (INCITE) allocations BIP117 and BIP115 and Summit Early Science Program (ESP) allocation BIP177. This work finally used resources from Blue Waters, which was supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the State of Illinois, through Petascale Computing Resource Allocations (PRAC) project BANV. The authors gratefully acknowledge helpful discussions with John Stone, Jim Phillips, Dave Hardy, Chris Mayne, and Josh Vermaas.

## 8 Author Contributions

ET conceived and supervised the study, and NT designed and executed it. NT prepared the initial draft of the manuscript, and ET and NT revised it together.

## 9 Competing Interests

The authors declare no competing interests.

## 10 References

- [1] P. L. Freddolino, A. S. Arkhipov, S. B. Larson, A. McPherson, and K. Schulten. “Molecular Dynamics Simulations of the Complete Satellite Tobacco Mosaic Virus.” *Structure*. **14**:3, 437–449 (2006). <https://doi.org/10.1016/j.str.2005.11.014>.
- [2] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, *et al.* “Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics.” *Nature*. **497**:7451, 643–646 (2013). <https://doi.org/10.1038/nature12162>.
- [3] T. Reddy, D. Shorthouse, D. L. Parton, E. Jefferys, P. W. Fowler, *et al.* “Nothing to Sneeze At: A Dynamic and Integrative Computational Model of an Influenza A Virion.” *Structure*. **23**:3, 584–597 (2015). <https://doi.org/10.1016/j.str.2014.12.019>.
- [4] J. M. A. Grime, J. F. Dama, B. K. Ganer-Pornillos, C. L. Woodward, G. J. Jensen, *et al.* “Coarse-grained simulation reveals key features of HIV-1 capsid self-assembly.” *Nature Communications*. **7**:1, 11568 (2016). <https://doi.org/10.1038/ncomms11568>.
- [5] J. K. Marzinek, D. A. Holdbrook, R. G. Huber, C. Verma, and P. J. Bond. “Pushing the Envelope: Dengue Viral Membrane Coaxed into Shape by Molecular Simulations.” *Structure*. **24**:8, 1410–1420 (2016). <https://doi.org/10.1016/j.str.2016.05.014>.
- [6] J. A. Hadden, J. R. Perilla, C. J. Schlicksup, B. Venkatakrishnan, A. Zlotnick, *et al.* “All-atom molecular dynamics of the HBV capsid reveals insights into biological function and cryo-EM resolution limits.” *eLife*. **7**, e32478 (2018). <https://doi.org/10.7554/eLife.32478>.
- [7] J. D. Durrant, S. E. Kochanek, L. Casalino, P. U. Jeong, A. C. Dommer, *et al.* “Mesoscale All-Atom Influenza Virus Simulations Suggest New Substrate Binding

- Mechanism.” *ACS Central Science*. **6**:2, 189–196 (2020). <https://doi.org/10.1021/acscentsci.9b01071>.
- [8] A. Yu, A. J. Pak, P. He, V. Monje-Galvan, L. Casalino, *et al.* “A multiscale coarse-grained model of the SARS-CoV-2 virion.” *Biophysical Journal*. **120**:6, 1097–1104 (2021). <https://doi.org/10.1016/j.bpj.2020.10.048>.
- [9] L. Casalino, A. C. Dommer, Z. Gaieb, E. P. Barros, T. Sztain, *et al.* “AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics.” *The International Journal of High Performance Computing Applications*. **35**:5, 432–451 (2021). <https://doi.org/10.1177/10943420211006452>.
- [10] Z. Ghaemi, M. Gruebele, and E. Tajkhorshid. “Molecular mechanism of capsid disassembly in hepatitis B virus.” *Proceedings of the National Academy of Sciences*. **118**:36, e2102530118 (2021). <https://doi.org/10.1073/pnas.2102530118>.
- [11] A. Singharoy, C. Maffeo, K. H. Delgado-Magnero, D. J. K. Swainsbury, M. Sener, *et al.* “Atoms to Phenotypes: Molecular Design Principles of Cellular Energy Metabolism.” *Cell*. **179**:5, 1098–1111 (2019). <https://doi.org/10.1016/j.cell.2019.10.021>.
- [12] I. Yu, T. Mori, T. Ando, R. Harada, J. Jung, *et al.* “Biomolecular interactions modulate macromolecular structure and dynamics in atomistic model of a bacterial cytoplasm.” *eLife*. **5**, e19274 (2016). <https://doi.org/10.7554/eLife.19274>.
- [13] J. Jung, W. Nishima, M. Daniels, G. Bascom, C. Kobayashi, *et al.* “Scaling molecular dynamics beyond 100,000 processor cores for large-scale biophysical simulations.” *Journal of Computational Chemistry*. **40**:21, 1919–1930 (2019). <https://doi.org/10.1002/jcc.25840>.
- [14] J. V. Vermaas, C. G. Mayne, E. Shinn, and E. Tajkhorshid. “Assembly and Analysis of Cell-Scale Membrane Envelopes.” *Journal of Chemical Information and Modeling*. **62**:3, 602–617 (2022). <https://doi.org/10.1021/acs.jcim.1c01050>.

- [15] J. A. Stevens, F. Grünewald, P. A. M. van Tilburg, M. König, B. R. Gilbert, *et al.* “Molecular dynamics simulation of an entire cell.” *Frontiers in Chemistry*. **11**, 1106495 (2023). <https://doi.org/10.3389/fchem.2023.1106495>.
- [16] M. Terasaki, T. Shemesh, N. Kasthuri, R. W. Klemm, R. Schalek, *et al.* “Stacked Endoplasmic Reticulum Sheets Are Connected by Helicoidal Membrane Motifs.” *Cell*. **154**:2, 285–296 (2013). <https://doi.org/10.1016/j.cell.2013.06.031>.
- [17] G. S. Tomassy, D. R. Berger, H.-H. Chen, N. Kasthuri, K. J. Hayworth, *et al.* “Distinct Profiles of Myelin Distribution Along Single Axons of Pyramidal Neurons in the Neocortex.” *Science*. **344**:6181, 319–324 (2014). <https://doi.org/10.1126/science.1249766>.
- [18] G. C. Garcia, T. M. Bartol, S. Phan, E. A. Bushong, G. Perkins, *et al.* “Mitochondrial morphology provides a mechanism for energy buffering at synapses.” *Scientific Reports*. **9**:1, 18306 (2019). <https://doi.org/10.1038/s41598-019-54159-1>.
- [19] S. Jo, T. Kim, V. G. Iyer, and W. Im. “CHARMM-GUI: A web-based graphical user interface for CHARMM.” *Journal of Computational Chemistry*. **29**:11, 1859–1865 (2008). <https://doi.org/10.1002/jcc.20945>.
- [20] E. L. Wu, X. Cheng, S. Jo, H. Rui, K. C. Song, *et al.* “CHARMM-GUI Membrane Builder toward realistic biological membrane simulations.” *Journal of Computational Chemistry*. **35**:27, 1997–2004 (2014). <https://doi.org/10.1002/jcc.23702>.
- [21] S. Jo, J. B. Lim, J. B. Klauda, and W. Im. “CHARMM-GUI Membrane Builder for Mixed Bilayers and Its Application to Yeast Membranes.” *Biophysical Journal*. **97**:1, 50–58 (2009). <https://doi.org/10.1016/j.bpj.2009.04.013>.
- [22] S. Jo, T. Kim, and W. Im. “Automated Builder and Database of Protein/Membrane Complexes for Molecular Dynamics Simulations.” *PLOS ONE*. **2**:9, e880 (2007). <https://doi.org/10.1371/journal.pone.0000880>.

- [23] J. Lee, D. S. Patel, J. Stähle, S.-J. Park, N. R. Kern, *et al.* “CHARMM-GUI *Membrane Builder* for Complex Biological Membrane Simulations with Glycolipids and Lipoglycans.” *Journal of Chemical Theory and Computation*. **15**:1, 775–786 (2019). <https://doi.org/10.1021/acs.jctc.8b01066>.
- [24] S. Park, Y. K. Choi, S. Kim, J. Lee, and W. Im. “CHARMM-GUI Membrane Builder for Lipid Nanoparticles with Ionizable Cationic Lipids and PEGylated Lipids.” *Journal of Chemical Information and Modeling*. **61**:10, 5192–5202 (2021). <https://doi.org/10.1021/acs.jcim.1c00770>.
- [25] T. A. Wassenaar, H. I. Ingólfsson, R. A. Böckmann, D. P. Tieleman, and S. J. Marrink. “Computational Lipidomics with *insane*: A Versatile Tool for Generating Custom Membranes for Molecular Simulations.” *Journal of Chemical Theory and Computation*. **11**:5, 2144–2155 (2015). <https://doi.org/10.1021/acs.jctc.5b00209>.
- [26] S. Schott-Verdugo and H. Gohlke. “PACKMOL-Memgen: A Simple-To-Use, Generalized Workflow for Membrane-Protein-Lipid-Bilayer System Building.” *Journal of Chemical Information and Modeling*. **59**:6, 2522–2528 (2019). <https://doi.org/10.1021/acs.jcim.9b00269>.
- [27] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez. “PACKMOL: A package for building initial configurations for molecular dynamics simulations.” *Journal of Computational Chemistry*. **30**:13, 2157–2164 (2009). <https://doi.org/10.1002/jcc.21224>.
- [28] K. J. Boyd and E. R. May. “BUMPY: A Model-Independent Tool for Constructing Lipid Bilayers of Varying Curvature and Composition.” *Journal of Chemical Theory and Computation*. **14**:12, 6642–6652 (2018). <https://doi.org/10.1021/acs.jctc.8b00765>.

- [29] G. T. Johnson, L. Autin, M. Al-Alusi, D. S. Goodsell, M. F. Sanner, *et al.* “cellPACK: a virtual mesoscope to model and visualize structural systems biology.” *Nature Methods*. **12**:1, 85–91 (2015). <https://doi.org/10.1038/nmeth.3204>.
- [30] J. D. Durrant and R. E. Amaro. “LipidWrapper: An Algorithm for Generating Large-Scale Membrane Models of Arbitrary Geometry.” *PLOS Computational Biology*. **10**:7, e1003720 (2014). <https://doi.org/10.1371/journal.pcbi.1003720>.
- [31] W. Pezeshkian, M. König, T. A. Wassenaar, and S. J. Marrink. “Backmapping triangulated surfaces to coarse-grained membrane models.” *Nature Communications*. **11**:1, 2296 (2020). <https://doi.org/10.1038/s41467-020-16094-y>.
- [32] T. W. Keenan and C. M. Huang. “Membranes of Mammary Gland. VI. Lipid and Protein Composition of Golgi Apparatus and Rough Endoplasmic Reticulum from Bovine Mammary Gland.” *Journal of Dairy Science*. **55**:11, 1586–1596 (1972). [https://doi.org/10.3168/jds.S0022-0302\(72\)85725-4](https://doi.org/10.3168/jds.S0022-0302(72)85725-4).
- [33] W. Humphrey, A. Dalke, and K. Schulten. “VMD: Visual molecular dynamics.” *Journal of Molecular Graphics*. **14**:1, 33–38 (1996). [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5).
- [34] J. C. Phillips, D. J. Hardy, J. D. C. Maia, J. E. Stone, J. V. Ribeiro, *et al.* “Scalable molecular dynamics on CPU and GPU architectures with NAMD.” *The Journal of Chemical Physics*. **153**:4, 044130 (2020). <https://doi.org/10.1063/5.0014475>.
- [35] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, *et al.* “MeshLab: an Open-Source Mesh Processing Tool.” *Eurographics Italian Chapter Conference* (2008). <https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136>.
- [36] J. B. Klauda, R. M. Venable, J. A. Freites, J. W. O’Connor, D. J. Tobias, *et al.* “Update of the CHARMM All-Atom Additive Force Field for Lipids: Validation on Six



- Lipid Types.” *The Journal of Physical Chemistry B*. **114**:23, 7830–7843 (2010). <https://doi.org/10.1021/jp101759q>.
- [37] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. “Comparison of simple potential functions for simulating liquid water.” *The Journal of Chemical Physics*. **79**:2, 926–935 (1983). <https://doi.org/10.1063/1.445869>.
- [38] D. Beglov and B. Roux. “Finite representation of an infinite bulk system: Solvent boundary potential for computer simulations.” *The Journal of Chemical Physics*. **100**:12, 9050–9063 (1994). <https://doi.org/10.1063/1.466711>.
- [39] R. M. Venable, Y. Luo, K. Gawrisch, B. Roux, and R. W. Pastor. “Simulations of Anionic Lipid Membranes: Development of Interaction-Specific Ion Parameters and Validation Using NMR Data.” *The Journal of Physical Chemistry B*. **117**:35, 10183–10192 (2013). <https://doi.org/10.1021/jp401512z>.
- [40] D. B. Wells, V. Abramkina, and A. Aksimentiev. “Exploring transmembrane transport through  $\alpha$ -hemolysin with grid-steered molecular dynamics.” *The Journal of Chemical Physics*. **127**:12, 125101 (2007). <https://doi.org/10.1063/1.2770738>.
- [41] T. Möller and B. Trumbore. “Fast, Minimum Storage Ray-Triangle Intersection.” *Journal of Graphics Tools*. **2**:1, 21–28 (1997). <https://doi.org/10.1080/10867651.1997.10487468>.
- [42] J. V. Vermaas, N. Trebesch, C. G. Mayne, S. Thangapandian, M. Shekhar, *et al.* “Microscopic Characterization of Membrane Transporter Function by In Silico Modeling and Simulation.” *Methods in Enzymology*. **578**:16, 373–428 (2016). <https://doi.org/10.1016/bs.mie.2016.05.042>.
- [43] D. H. de Jong, G. Singh, W. F. D. Bennett, C. Arnarez, T. A. Wassenaar, *et al.* “Improved Parameters for the Martini Coarse-Grained Protein Force Field.” *Journal of*

- Chemical Theory and Computation*. **9**:1, 687–697 (2013). <https://doi.org/10.1021/ct300646g>.
- [44] T. A. Wassenaar, K. Pluhackova, R. A. Böckmann, S. J. Marrink, and D. P. Tieleman. “Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models.” *Journal of Chemical Theory and Computation*. **10**:2, 676–690 (2014). <https://doi.org/10.1021/ct400617g>.
- [45] P. C. T. Souza, R. Alessandri, J. Barnoud, S. Thallmair, I. Faustino, *et al.* “Martini 3: a general purpose force field for coarse-grained molecular dynamics.” *Nature Methods*. **18**:4, 382–388 (2021). <https://doi.org/10.1038/s41592-021-01098-3>.
- [46] G. J. Martyna, D. J. Tobias, and M. L. Klein. “Constant pressure molecular dynamics algorithms.” *The Journal of Chemical Physics*. **101**:5, 4177–4189 (1994). <https://doi.org/10.1063/1.467468>.
- [47] S. E. Feller, Y. Zhang, R. W. Pastor, and B. R. Brooks. “Constant pressure molecular dynamics simulation: The Langevin piston method.” *The Journal of Chemical Physics*. **103**:11, 4613–4621 (1995). <https://doi.org/10.1063/1.470648>.
- [48] T. Darden, D. York, and L. Pedersen. “Particle mesh Ewald: An N·log(N) method for Ewald sums in large systems.” *The Journal of Chemical Physics*. **98**:12, 10089–10092 (1993). <https://doi.org/10.1063/1.464397>.
- [49] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, *et al.* “A smooth particle mesh Ewald method.” *The Journal of Chemical Physics*. **103**:19, 8577–8593 (1995). <https://doi.org/10.1063/1.470117>.