TS2CG as a membrane builder

Fabian Schuhmann[1,*], Jan A. Stevens[2,*], Neda Rahmani[1], Isabell Lindahl[1], Chelsea M. Brown[2], Christopher Brasnett[2], Dimitrios Anastasiou[1], Adrià Bravo Vidal[1], Beatrice Geiger[1], Siewert J. Marrink[2], and Weria Pezeshkian[1,†]

[1] *Niels Bohr International Academy, Niels Bohr Institute, University of Copenhagen, Blegdamsvej 17, 2100 Copenhagen, Denmark*

[2] *Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen, Nijenborgh 7, 9747 AG Groningen, Netherlands*

[*] *Equal Contribution*

[†] *weria.pezeshkian@nbi.ku.dk*

July 21, 2025

Abstract

Molecular dynamics (MD) simulations excel at capturing biological processes at the molecular scale but rely on a well-defined initial structure. As MD simulations now extend to whole-cell-level modeling, new tools are needed to efficiently build initial structures. Here, we introduce TS2CG version 2, designed to construct coarse-grained membrane structures with any desired shape and lateral organization. This version enables precise placement of lipids and proteins based on curvature preference, facilitating the creation of large, near-equilibrium membranes. Additional features include controlled pore generation and the placement of specific lipids at membrane edges for stabilization. Moreover, a Python interface allows users to extend functionality while maintaining the high performance of the C++ core. To demonstrate its capabilities, we showcase

challenging simulations, including a Möbius strip membrane, a vesicle with lipid domain as continental plates (Martini globe), and entire mitochondrial membranes exhibiting lipid heterogeneity due to curvature, along with a comprehensive set of tutorials.

Introduction

Molecular dynamics (MD) simulations have emerged as a powerful and indispensable tool for exploring biological processes at the molecular scale. Advances in force field development [1–4], simulation engines [5–7], and computing hardware have expanded the scope of MD simulations, from studying individual proteins to complex molecular systems. A frontier direction in the biomolecular modeling field is the simulation of entire cells or cell organelles with molecular resolution, often referred to as in situ simulations [8]. While traditional approaches study isolated biological components, in situ simulations preserve realistic complex molecular interactions similar to their native cellular environment, allowing for synergetic effects to emerge. Achieving such comprehensive simulations might have seemed impossible two decades ago. However, successful simulations of cellular systems, including reduced minimal cells and mitochondrial membranes, among others, have demonstrated the feasibility of this goal [9–13]. Now, that these pioneering achievements have been accomplished, it is time to streamline the process and to broaden the usage of the MD simulations potential. For the simulation of small molecular systems, starting from a random configuration of single lipid-like molecules often suffices to achieve a converged equilibrium state within a feasible simulation time. This is well-established through classic studies, such as lipid bilayer self-assembly within hundreds of nanoseconds [14]. However, constructing initial configurations for cell scale simulations using traditional tools, if even possible, is daunting, time-consuming, and requires many preparation steps. Therefore, a new generation of tools is needed, ones that leverage experimental data and theoretical insights to better initialize simulations so that these can capture the complexity of cellular environments [13, 15–18].

A major bottleneck in modeling large-scale biological systems is the cellular membrane, which exhibits a highly complex organization that emerges over timescales well beyond the feasible timescale of the current MD models. For example, membrane proteins form functional clusters while attracting characteristic lipid shells, and both can separate into distinct domains based on their molecular properties. These organizational patterns are also coupled with the membrane shape, as both proteins and lipids have their own preferred local curvature [19–21]. While MD simulations can potentially capture the formation of these higher-order structures, they require extensive computational resources to reach equilibrium. This becomes particularly challenging in the context of large in-situ models, where multiple such processes would need to occur simultaneously and the required timescale increases nonlinearly with the system size. To study such systems effectively, we must construct initial states that reflect their natural organization as closely as possible by integrating prior knowledge from both experimental and modeling studies. The complexity is also influenced by the membrane shape, which has been discussed in an earlier study [13]. Earlier tools for membrane structure generation primarily focused on circumventing the self-assembly process by pre-assembling lipids into a bilayer configuration. They can generate lipid bilayers in standard shapes, such as flat membranes and vesicles, with a user-defined lipid composition but random lateral organization. Additionally, they often allow the integration of a limited number of proteins [17, 22, 23]. In recent years, a separate class of tools has emerged that can handle arbitrary membrane shapes [24–26]. However, the tools only allow membrane creations according to certain predefined puzzle shapes, do not consider the lateral organization of lipids, require an already generated flat membrane as an input, or cannot place proteins inside or around the membrane. [24–26]. For this purpose, we introduce TS2CG2.0 (**T**riangulated **S**urface to(**2**) **C**oarse **G**rained) as a membrane builder. The program stands out, allowing the design of membrane systems with arbitrary shapes and customizable lipid and protein compositions [13]. Initially, TS2CG was developed for backmapping dynamically triangulated simulation [27, 28] structures into their corresponding coarse-grained (CG) molecular models as an important element for multiscale

membrane simulations using, by default, the Martini force field. As TS2CG focuses on the structural placement, the program can straight-forwardly be adapted to handle lipids and structures described in arbitrary forcefields. Furthermore, TS2CG emerged as a robust tool to incorporate experimentally obtained membrane shapes and compositions into the initial configuration of a CG membrane simulation. Yet, a key challenge remains: generating near-equilibrium lateral organization, especially when accounting for the interplay between membrane shape and molecular composition [29–31]. This feature is particularly crucial for modeling the highly curved shapes of organelles, where the membrane shape strongly affects protein and lipid organization and vice versa [9, 29]. TS2CG 2.0 enables precise control over lateral molecular organization during model construction and can be supplied with an arbitrary membrane geometry through triangulated meshes. The new features could enable the generation of more realistic membrane models by incorporating experimental constraints directly into the building process and thus aligning in-silico studies directly with their experimental counterparts.

While we report on case studies using the Martini 3 force field, TS2CG 2.0 has also been successfully used to generate ready to simulate membranes for the Sirah [2] and CHARMM [32] force fields in two specific simple cases (see S6 Non-Martini membranes). However, we leave testing on more complex systems for future studies or interested users.

Results

Building Membranes

Building a membrane ready for simulation via TS2CG 2.0 is done in multiple steps, combining different tools and ideas. Simple membranes, however, can now be generated in a single step by utilizing the analytical shape feature of step 3. The program is hosted on GitHub at `https://github.com/weria-pezeshkian/TS2CG-v2.0/`. Additional documentation can

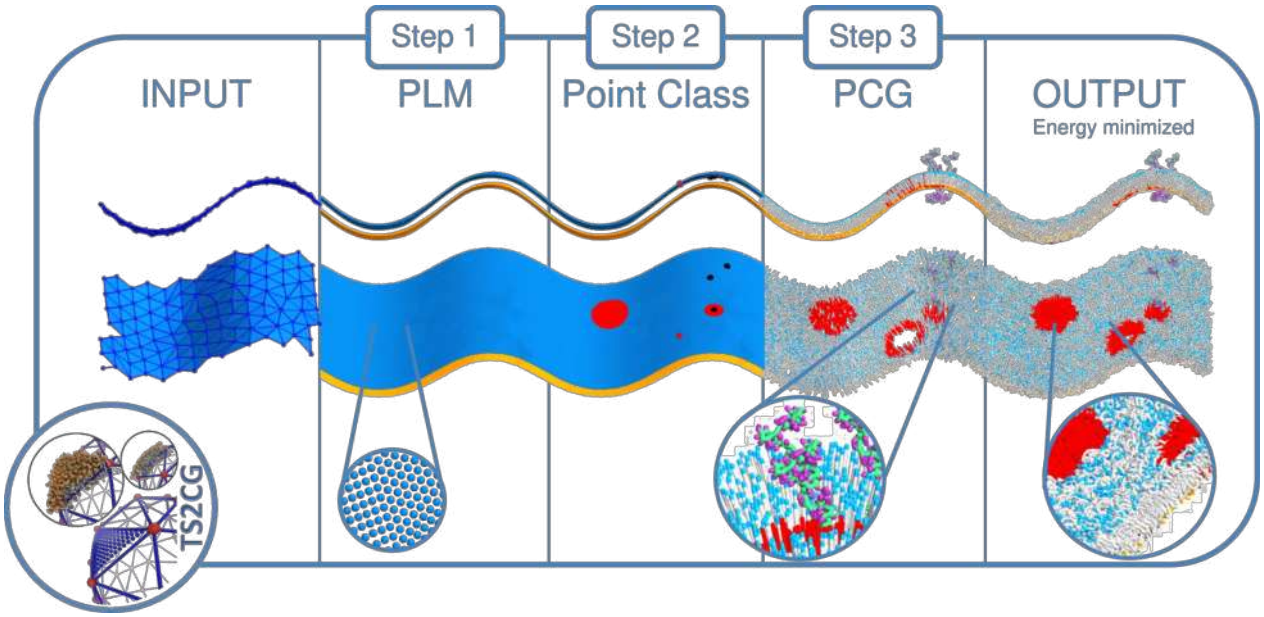

Figure 1: The TS2CG 2.0 membrane creation workflow A workflow diagram explaining the overall steps of TS2CG 2.0 to generate complex membranes. The process can be started via an analytical shape or an arbitrary triangulated surface. Here, we consider a sin shape. Through PLM or PCG, a point directory can be created which is then manipulated using the Point class to place proteins, exclusions or introduce a different domain (inclusions (proteins) as black dots, domain in orange, and the center of an exclusion is marked in red). A second execution of PCG turns the point folder including the changes into a membrane structure ready for subsequent simulation. PLM and PCG are names for TS2CG 2.0 subroutines, which are explained in the step-by-step guide.

be found at https://weria-pezeshkian.github.io/TS2CG_python_documentation/. The details of the different procedures will be discussed below in the methods section.

Step 0 Installation

TS2CG 2.0 is distributed through PyPI, enabling installation via pip. While the software's computationally intensive operations are implemented in C++, PyPI handles the compilation of these binary components automatically. Users interact with TS2CG 2.0 through Python and command-line interfaces, combining the accessibility of Python with the performance benefits of compiled code.

Step 1 Surface Discretization

In the first step, the general shape of the to-be-built membrane is created. For complex membrane shapes, a triangulated mesh is converted into a discrete point representation through the PLM (pointillism) subroutine. The points inherit the attributes of their corresponding surface element, which include a surface area, local coordinate, and curvature tensor. They are generated using a specific algorithm introduced in the first version, which ensures the surface retains its curvature, a critical feature governing membrane behavior at larger scales. PLM accepts triangulated meshes in .tsi format, which can be extracted, for instance, from FreeDTS [28] simulation software output or generated through modeling software like Blender [33–35]. PLM converts the triangulated mesh surface into a discretized set of points and writes a point folder collecting the inner and outer membrane, as well as inclusions (an abstract for proteins location in/on the membrane) and exclusions (an abstract for holes in the membrane). A new extension to PLM is handling shapes that are not closed and allows special treatment of edge cases, see, for instance, the case study Open-edge geometries below. The open edges allow the creation of a different classes of membranes such as nano disks. The points represent surface elements that have spatial coordinates r , area, normal vectors $\hat{n}$, and principal curvatures as well as principal directions. The generated point distribution serves as a template for placing molecules during membrane construction while preserving the original surface geometry.

PLM thus transfers a triangle mesh into the point folder, which can be read by the membrane building subroutine (PCG) to construct the membrane structures.

For simple membranes, TS2CG also supports the creation of membranes and direct placement of lipids. A set of analytical geometries are supported, i.e., flat membranes, sine and cosine-shaped membranes, vesicles, or cylinders. This straightforward generation of membranes is implemented in PCG. Additionally, for these geometries, a point directory can be created, which can be used to further modify the membrane composition, as described later.

Step 2 Proteins, Pores, and Lipid Domains

TS2CG 2.0 incorporates a Python-based interface for manipulating the generated points. This framework enables precise control over membrane properties through three primary components: membrane composition, Inclusions (protein placement), and Exclusions (membrane pores, which can be introduced to study hole formation and closure). The Point class API allows direct programmatic access for analyzing and modifying membrane composition, enabling informed placement of molecules based on characteristics such as curvature or any user-defined geometric parameter, once the manipulations are handed off to PCG. In case the specific distribution of the inclusions, exclusions, or lipids is not specified, the routine falls back to a random placement. To put the framework into practice, we provide command-line tools that implement common membrane modification strategies, such as creating circular lipid domains around proteins or around any arbitrary point (DAI), assigning lipids based on curvature preference (DOP), or inserting membrane proteins based on curvature preference (INU). These tools enable users to integrate prior knowledge into the membrane composition without the need for Python expertise. The routines are described in detail below.

DAI: ("**D**omains **A**round **I**nclusions') Creates circular domains of a user-defined radius around protein inclusions or arbitrary points. The basic method directly assigns domains based on the Euclidean distance from a user-specified central point. Although this is an efficient implementation, it can produce artifacts if a membrane curves so much, that the different regions of the membrane come closer to each other than the assigned domain radius (see Figure 5 for an example). To resolve this issue, a more computationally intensive, graph-based algorithm is also implemented that employs Dijkstra's shortest path algorithm [36] to ensure domain continuity within individual membrane surfaces. This latter method prevents artifacts in multi-membrane systems by constructing a distance-weighted network based on the underlying mesh of the distributed points. This tool enables precise control of the membrane composition regarding protein-lipid interactions and the organization of lipid domains. The

user chooses which implementation is used when running the program. DAI can assign one domain type per central point but can be run multiple times to add multiple different domains. However, in the case of an overlap to a previous run, the overlapping area will be completely dominated by the last execution of DAI. An application of the procedure is provided in the case study entitled Mitochondrial membrane.

DOP: ('**D**istribution-based **O**ptimized **P**lacement') Assigns lipid compositions based on the membrane geometry using the local mean curvature. The placement algorithm iterates over each point in the discretized membrane in random order to prevent a systematic bias. For each point, the placement probability for a given lipid type (l) is calculated based on the local mean curvature (H) and the lipid's intrinsic curvature preference (C_0) using a Boltzmann-weighted function,

$$P(l) = e^{-k(2H-C_0)^2}, \tag{1}$$

where k is a user-defined scaling factor controlling domain sharpness. Additionally, a user can scale k further by the area of each point. Through this scaling, the Eq. (1) resembles the Helfrich Hamiltonian. These probabilities, normalized across all lipid types $\left(P(\hat{l})/\sum_{l\in[lipid\ types]} P(l)\right)$, guide the stochastic lipid assignment while preserving the specified overall lipid composition. The curvature-informed placement of lipids allows for the building of membranes closer to equilibrium, substantially reducing the computational cost of the subsequent MD equilibration simulation. The procedure is applied in the case study Integrating Lipid Curvature Preference. It is noted that DOP provides only the weighted function above; however, arbitrary functions are conceivable. In principle, the Point Class API enables complete customization of the probability function used, allowing the implementation of arbitrary lipid sorting schemes tailored to a specific research question.

INU: ('**IN**clusion **U**pdater') Facilitates easy collision-free protein placement based on local membrane curvature, using the same probabilistic assignment as DOP. For each potential protein position, a placement probability is calculated over the available membrane points

using the local mean curvature H, the protein's preferred curvature C_0, and the user-defined scaling factor k that controls specificity. Protein-protein collisions are prevented through a user-defined collision radius. This approach enables the generation of biologically relevant protein distributions across complex membrane geometries. The procedure is used in the case study Mitochondrial membrane.

While these pre-built tools allow useful applications, the underlying Point class API gives users complete programmatic control to develop custom point modification schemes tailored to their specific membrane modeling requirements. Also note, that for most applications, the subroutines through the Python API manipulate the point class and do not generate the membrane. Therefore, execution of the subroutines is significantly faster than the membrane builder described below. Detailed documentation of the Point class implementation is available in Supplementary Material S3 Python documentation.

Step 3 Membrane building

The final step in the membrane assembly entails the placement of the lipids and proteins based on the point distribution. The PCG subroutine achieves an overlap-free initial configuration of the membrane system by iteratively placing membrane components according to their spatial requirements, maintaining local geometric relationships and respecting the specified areas per lipid types. A membrane structure (.gro) file accompanied by the respective topology (.top) is returned, ready for simulation with the Gromacs simulation engine [37]. TS2CG, by default, relies on the Martini 3 force field and includes a file (Martini3.LIB) to place most Martini3 lipids. This file can be freely adjusted or replaced to allow placing arbitrary beads, which can then be made meaningful with a custom forcefield which will have to be included in the topology file. A helper program to generate a library entry from a single lipid structure file has been included; *libmaker*. TS2CG 2.0 has been executed successfully with Martini 2 (Martini Globe) and a POPC membrane employing SIRAH beads.

Additionally, given a provided protein structure, PCG can directly place proteins in the membrane. The precise location can be specified through the Point Class API, INU, or a random placement can be achieved. For each protein the z-height can be provided in the configuration input file to have the structure inside the membrane at the right depth or have it hover above or below. With this, it can be adjusted how far a protein might stick out of the membrane or how far it floats above the membrane at the start of the simulation to directly control for physiological conditions. The proteins are placed directly as specified in the input file and no post-placement relaxation is performed. Additionally, a pre-oriented protein structure yields better results as the placement height in z direction is determined from the z direction in the protein structure file. If the specified height is 0, the protein's center will be in the center of the membrane. Furthermore, if a protein exists in multiple orientations, it can be redefined as multiple proteins in their respective orientation with a separate structure file for each configuration allowing full user control.

Note, that the membranes or membrane protein complexes generated by TS2CG 2.0 serve as an initial structure that ought to be minimized and equilibrated to achieve a realistic membrane (within the bounds of the forcefield).

TS2CG 2.0 also includes practical utilities to simplify the membrane-building workflow. A visualization tool (VIS) enables easy visualization of the point distributions via the command line, allowing validation of domain assignments before membrane construction. Furthermore, the package also provides a solvation tool. This tool works similarly to gmx solvate [37] but is specifically designed for the fast propagation of a small, equilibrated water box into a larger box containing certain particles. It ensures that solvent particles do not overlap with existing particles within a user-defined cutoff radius and also allows for fast placement of ions. A comprehensive tutorial series demonstrating these features and core functionalities is available in the supplementary materials and on the GitHub repository.

Case Studies

To demonstrate the versatility of TS2CG 2.0, we constructed several membrane systems, some biologically relevant and others that showcase the technical limits of the program's new features. For each system, we performed MD simulations to validate that the created initial structure is simulation-ready. While the first example highlights the simpler functions of the daily routine of membrane-protein system initialization, the other examples emphasize TS2CG 2.0's ability to handle arbitrary membrane shapes, complex protein-lipid organizations, and non-standard membrane compositions. An extensive tutorial containing vesicles and flat membranes can be found in the supplementary material (S1 Tutorials).

A flat membrane

While the later case studies focus on the more specialized features of TS2CG 2.0, this first case study aims to showcase the creation of a flat membrane with two lipids and a single protein placed slightly above the membrane. For the creation of the presented system, elements from Tutorial 4 and 6 were combined as presented in the supplementary information (S1).

The membrane contains two types of lipids. One type has an occurrence of 90% and an area per lipid of 0.64 nm^2. The other makes up 10% of the lipids with an area per lipid of 0.77 nm^2. The system is amended by a single protein structure, which is placed randomly on the membrane with an elevation of 5 nm to the membrane. The thus created initial system was created in 0.87 seconds and is shown in Fig. 2A.

After a short minimization and equilibration a short simulation was performed akin to the standard simulation setup described in the supplementary methods and the tutorials. The resulting structure (excluding solvent) is visualized Fig. 2B.

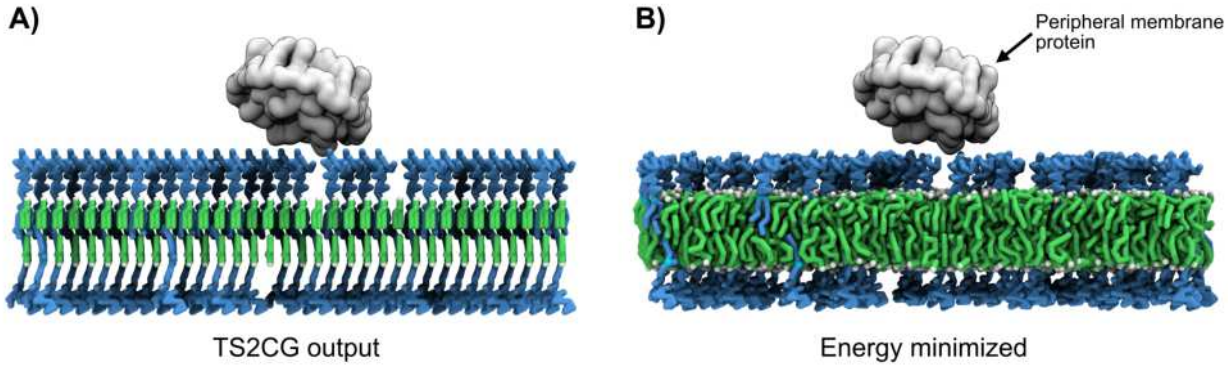

Figure 2: A flat membrane A membrane with a single protein is shown after initialization through TS2CG (A) followed by a subsequent short MD simulation (B). The image has been centered around the protein.

Integrating Lipid Curvature Preference

Here, different lipids are placed according to the curvature preference profile given in Eq. (1). We show the results for varying TS2CG parameters (k and C_0) to place around 20% cardiolipin (CDL) in a curved POPC membrane. The DOP feature to scale k further in Eq. (1) is not utilized for this case study. The C_0 value represents the lipid's curvature preference whereas the k-value is a scaling factor controlling domain sharpness. The aim is to generate membranes closer to equilibrium from the start. CDL is known to favor negatively curved regions [29]. Hence, we will work with fixed values of k and C_0 for POPC and test how varying k and C_0 values impact the placement of CDL. Nine different C_0 values were tested, each with ten different k-values, resulting in a total of 90 systems. All test systems underwent a brief energy minimization. We then compare the biased placement of TS2CG with the end result of a 1 μs simulation which allowed for the natural sorting of the lipids after an initial random placement of the different lipid types. The simulated self-sorted membrane is considered to be the reference membrane for this case study.

The results are analysed based on two different methods. First by calculating lipid-type community scores to quantify the impact of parameter choices in DOP on the finally generated membrane structure. A second analysis compares the CDL density of a single-parameter set

to the self-sorted reference simulation. For further information regarding the methods and scores, we refer to the supplementary methods.

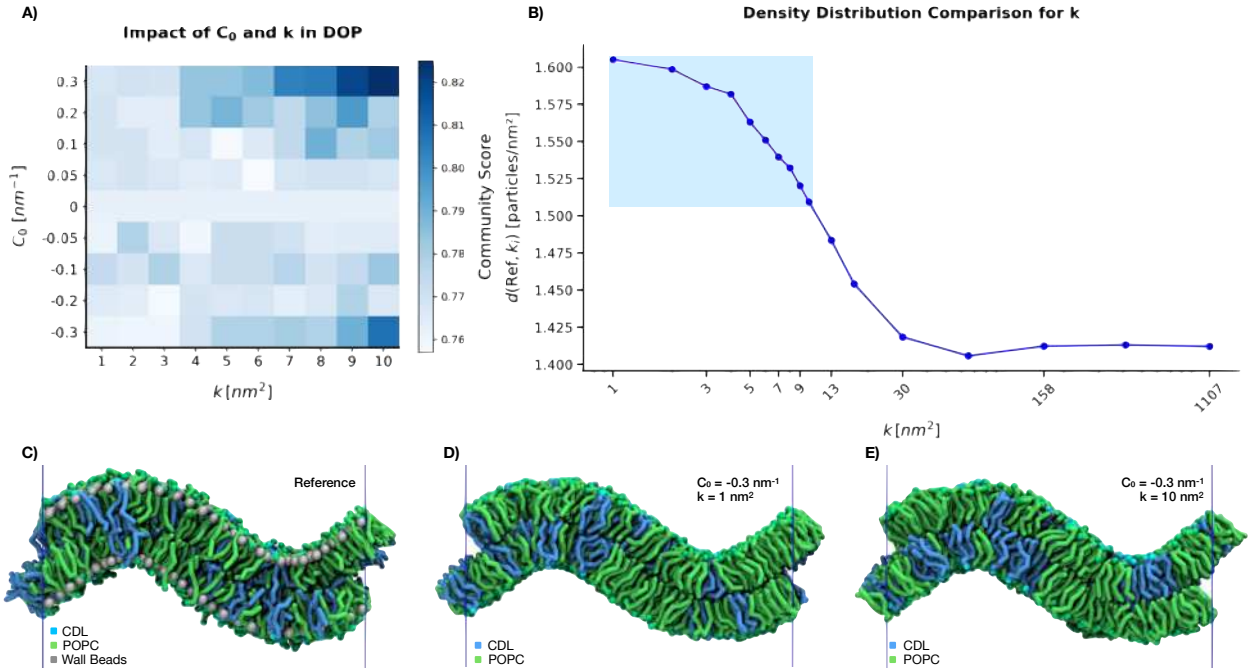

Figure 3: Generating membranes by sorted lipids according to their curvature preference A) Presents the community scores of the 90 membranes generated with DOP and visualizes the impact of varying C_0 and k values for CDL while C_0 and k remain constant for POPC ($C_0 = 0,\ k = 1$). B) Density distribution comparison for different k-values, twenty systems generated with $C_0 = -0.3$ and different k-values are evaluated against the reference. The blue box highlights the k-values used for the community score. C) The reference system, with natural sorting of the lipids through simulations. D) Resulting system for C_0 of -0.3 and k of 1. E) Resulting system for C_0 of -0.3 and k of 10. The differences between Panels D and E become best visible in the right valley. The high k placement of Panel E places a collection of CDL precisely, while the low k placement does not. The blue lines in Panels C to E resemble the periodic boundary condition.

The results from the community scores, are presented in Figure 3 (A), whereas the results from the density distributions comparison are presented in Figure 3 (B). In Figure 3 (A), higher scores indicate stronger internal connectivity within a community. Conversely, lower scores signify more evenly distributed lipids. Hence, the higher the score, the more clear-cut lipid domains are formed on the membrane and the clearer CDL has been sorted to a certain region on the membrane. Three key observations can be made from Figure 3 (A). First,

an increase in the parameter k, that controls the strength of the bias, results in higher scores. Second, both high and low values of the spontaneous lipid curvature, C_0, contribute to increased scores. As expected, in absence of spontaneous curvature ($C_0 = 0$), no significant communities are observed, regardless of the k value. This is because C_0 is also equal to 0 for POPC, meaning that for every point, both lipid types have the same probability of being assigned, making the assignment effectively random. This only holds true because both lipid types share the same C_0 value. Overall, higher k-values result in higher scores and a more segregated membrane. The highest community score is observed for the largest spontaneous curvature tested ($C_0 = 0.3$), in combination with the largest bias $k = 10$, while another high score is found for $C_0 = -0.3$, also with $k = 10$. For $C_0 = 0.3$, PCG places the CDL lipids in the positively curved regions, whereas for $C_0 = -0.3$, PCG places these lipids in negatively curved regions. The positively curved regions are larger than the negatively curved regions due to asymmetry of the membrane shape. As a result, more lipids can be placed in the positively curved areas, which increases the connectivity within communities, leading to the highest community score. Further analysis will focus on the effect of a higher k-value where 20 systems were created with the same C_0 value but different k-values. The density data of CDL for each system was extracted and treated as distributions. As shown in Figure 3 (B), the density distributions of these 20 systems were compared to the distribution of the reference system using the Wasserstein distance, where a shorter distance indicates a greater similarity between distributions [38]. The observed trend shows that increasing the k-value leads to a CDL density distribution that more closely resembles that of the reference system. It also shows that there is a limit of how far we bias the placement, the curved areas of the membrane get saturated and PCG cannot place more lipids in the region. The results demonstrate that incorporating DOP into TS2CG 2.0 enables the creation of system conformations closer to a pre-defined equilibrium without requiring long, computationally expensive simulations. While achieving the pre-sorted structures took only minutes on a laptop CPU, the reference structure required hours of equilibration on a high-performance

cluster with GPU acceleration.

Mitochondrion

To explore the integrative modeling strengths of TS2CG 2.0, we built two complementary mitochondrial models: a detailed crista junction based on earlier work by Brown et al. [9] with experimentally informed protein and lipid distributions and a large-scale inner mitochondrial membrane. The first showcases how experimental and simulation data can be integrated into a complex membrane model, while the second presents the software's capacity to handle mesoscale membrane geometries.

Membrane curvature plays a crucial role in organizing both lipids and membrane proteins within the mitochondrial inner membrane [39], maintaining its complex membrane geometry. In both models, lipids are placed according to experimentally and computationally determined curvature preferences [29]. Specifically, POPC and POPS were preferentially placed in areas of positive curvature ($C_0 = 1\ nm^{-1}$, $k = 250\ nm^2$), while SAPE and PAPI were placed in areas of low curvature ($C_0 = 0\ nm^{-1}$, $k = 250\ nm^2$), and CDL2 placement was biased towards negatively curved regions ($C_0 = -1\ nm^{-1}$, $k = 250\ nm^2$). The realized lipid distribution in the model of the inner mitochondrial membrane, built using an experimentally obtained membrane geometry [40], shows a strong correlation with the curvature of the input structure (Figure 5A,B). Analysis of the relative lipid distributions shows the expected enrichment patterns in accordance with each lipid's curvature preferences (Figure 4C). The generation process of the whole inner mitochondrial membrane took roughly ten minutes with TS2CG 2.0.

The mitochondrial membrane proteins are similarly organized based on curvature preference. In order to create a relevant in situ model of a mitochondrial crista, we position eight different types of inner protein complexes according to experimentally determined localization on a triangulated surface representing a crista junction [9]. The placement of each protein complex

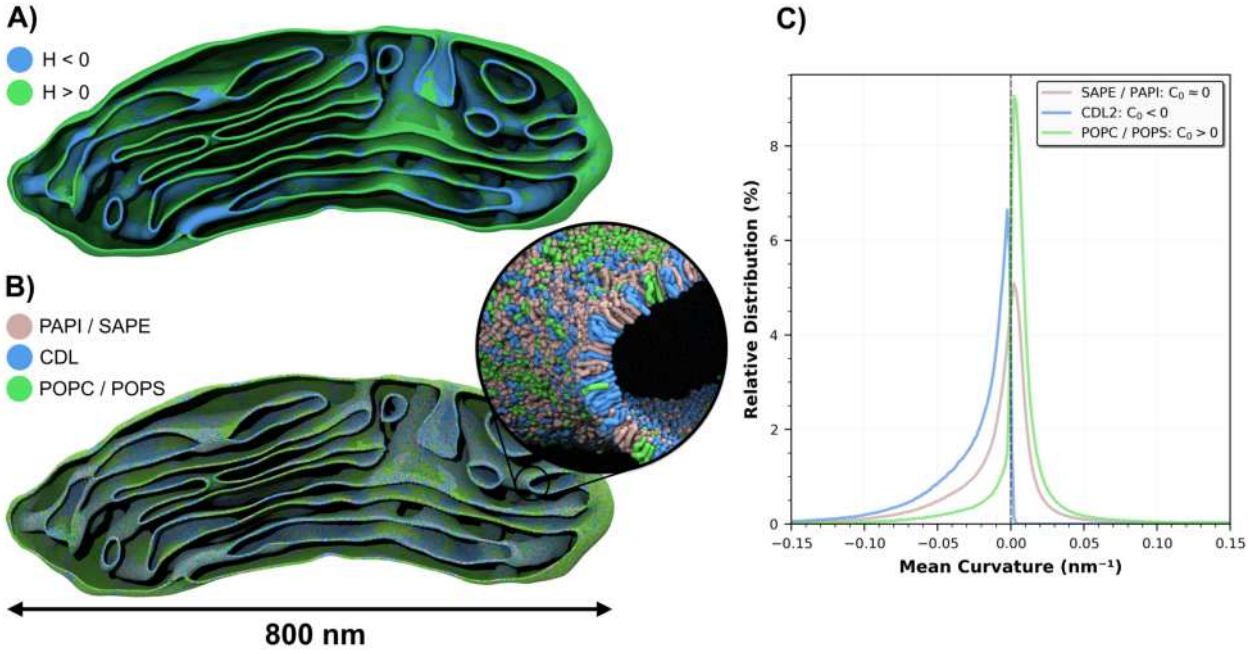

Figure 4: Mitochondrial membrane with real shape from Cryo-ET A) Inner mitochondrial membrane colored by mean curvature: positive mean curvature ($H > 0$) in blue, negative mean curvature ($H < 0$) in green. B) Martini 3 model of the inner mitochondrial membrane with lipids distributed by curvature preference (PAPI/SAPE pink, CDL blue, POPC/POPS green). C) Curvature-dependent distribution of mitochondrial lipids plotted as the relative distribution in function of membrane curvature. A large k was chosen to strongly bias the lipids for a clearer visualization of the tool. Therefore, the curves appear skewed as it is highly unlikely for a lipid to be placed in the region opposite its own preference.

is biased based on specific restrictions on both local mean curvature and z-position to represent its physiologically relevant localization.

As ATPase dimers are found at the crista ridge [41], these complexes were placed on vertices with negative curvature exclusively. The respiratory mega-complexes, ANT1 and ANT2 are found on the crista sides [42, 43] and hence were placed on flat regions of the membrane below the crista junction. The MIC60, TIM22, and TIM23 complexes are found on the inner membrane facing the outer mitochondrial membrane [44, 45], and hence these complexes were also placed on flat membrane regions but with a higher z-position restriction. The numbers of each protein included reflect concentrations reported in the literature [46].

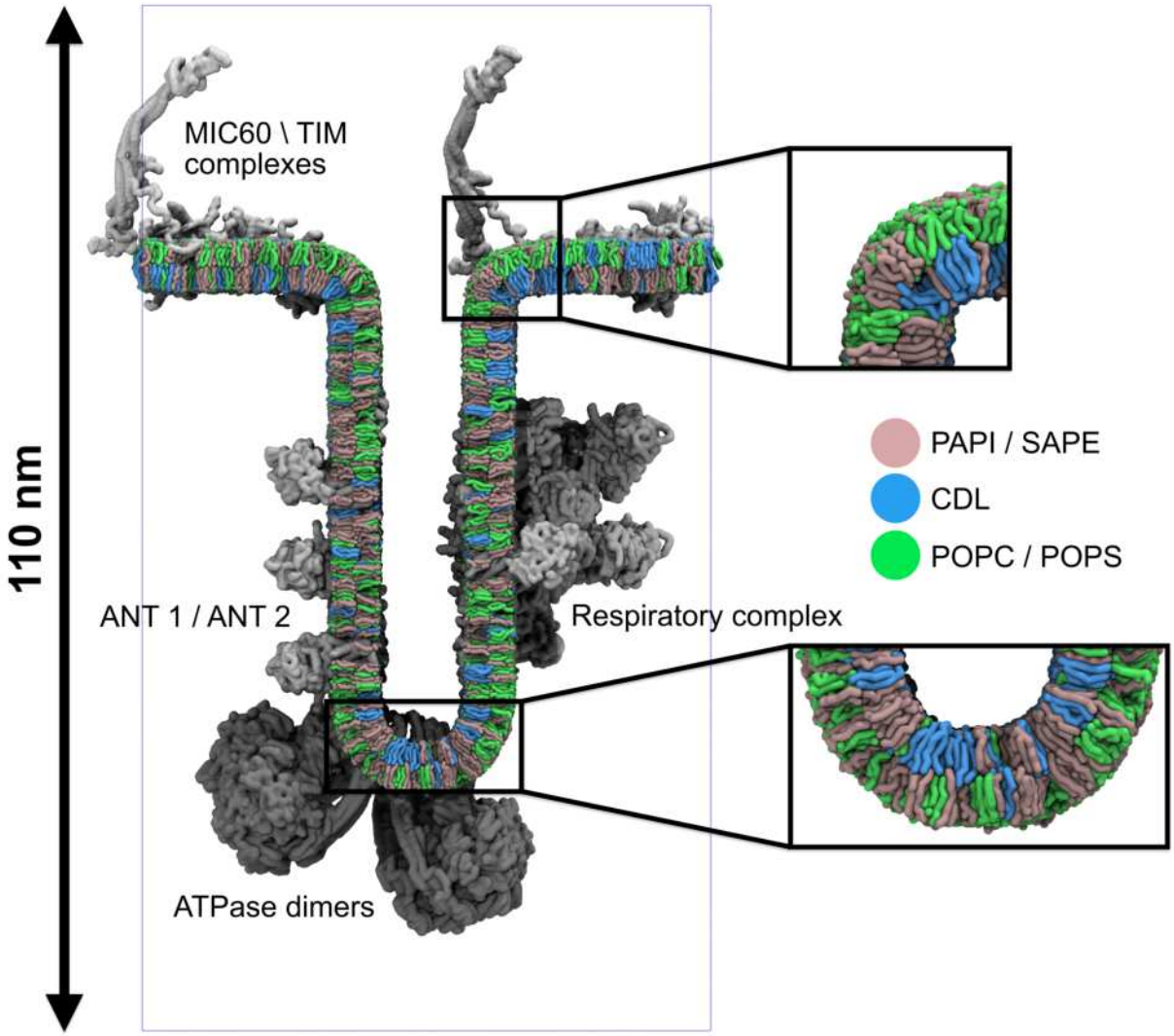

Figure 5: Model of a mitochondrial cristae The model shows lipids and membrane proteins arranged by curvature preference. MIC60/TIM complexes (top), ANT1/ANT2 transporters and respiratory complex (sides), and ATPase dimers (bottom) are shown. Zoom panels highlight the lipid sorting patterns: PAPI/SAPE (pink), CDL (purple), POPC/POPS (green).

The result of this procedure (Figure 5) is an accurate representation of a complex membrane system, showing curvature-based organization of both protein complexes and lipids based on curvature and membrane position. This enables the construction of complex systems with components placed in such a way that reflects those found in vivo, reducing the amount of simulation time needed to obtain meaningful results.

Open-edge Geometries

TS2CG 2.0 introduces support for open-edge meshes, further expanding its scope of application. Supporting open edges enables the construction of both biologically relevant systems (e.g. nanodiscs) and abstract geometries that were previously difficult or impossible to build.

To illustrate the power of this capability, we present a Möbius strip lipid membrane. The Möbius strip presents a challenging geometry as a non-orientable surface with a single continuous edge. The non-orientable nature results in a discontinuity in the surface normals across the mesh. As such an input mesh will always be cut through to recreate the orientable surface, and thus leading to open edges in all directions. TS2CG 2.0 can now incorporate this open-edge geometry and produce a Möbius strip membrane.

We constructed two versions of this Möbius membrane to showcase the edge-based lipid placement available in the new Python API (Figure 6). The first membrane consists of a simple binary mixture of 30% cholesterol and 70% POPC lipids, randomly placed. When simulated, the membrane's line tension drives a reduction in edge length, with the system evolving from the initial Möbius strip geometry to an intermediate Sudanese Möbius configuration (where the edge forms a circle, minimizing line tension energy) and finally to an oblate vesicle over a few hundred nanoseconds of CG simulation time. For the second version, we identified edge vertices in our mesh and selectively placed specific lipids at these locations. Since shorter lipids might stabilize the high curvature at the membrane edge, we placed DLPC lipids specifically along the edge vertices. The two systems showed different kinetics, although both eventually formed oblate vesicles. While the POPC/CHOL system completed its transition by 300 ns, the addition of DLPC noticeably delayed this process by a 100ns. Of course, to validate the statement statistical replicas would need to be conducted, which is beyond the scope of this proof-of-concept. For quantitative analysis of the topological transitions using leaflet segmentation, see Supplementary Figure S1. These simulations serve primarily as proof of concept for the technical capabilities of TS2CG 2.0, illustrating how programmatic

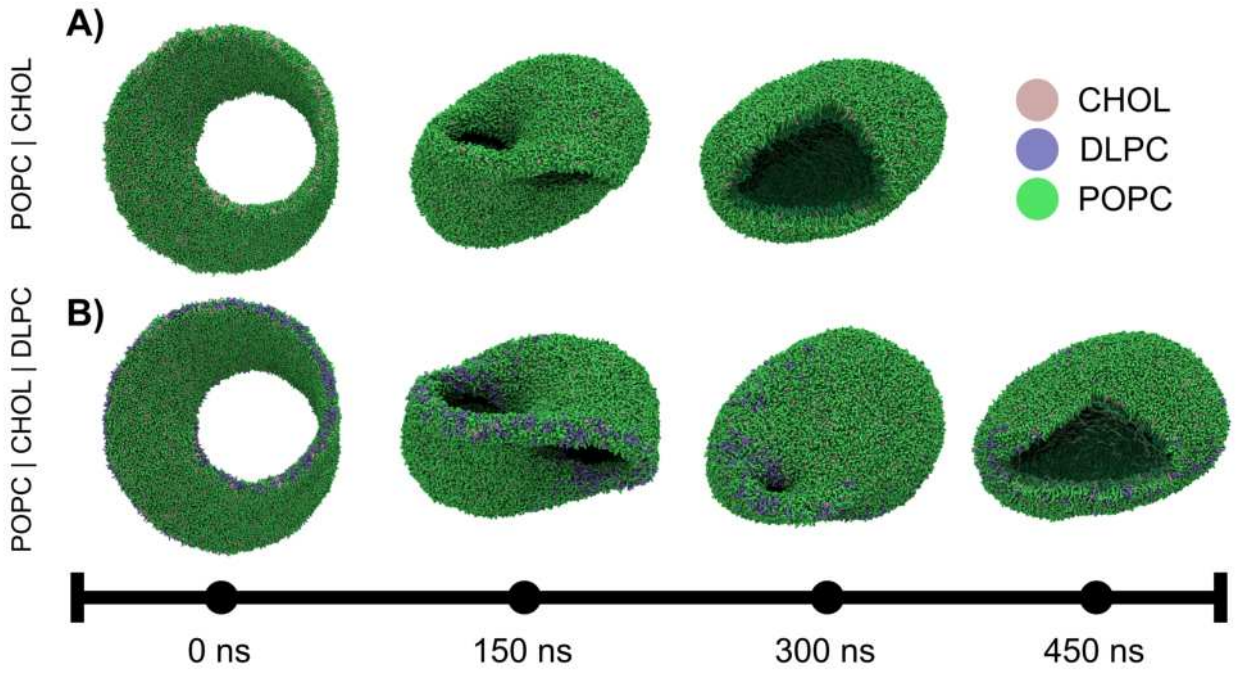

Figure 6: The Martini Möbius strip Representative snapshots of lipid Möbius strips transforming into vesicles. A) 70%POPC /30%CHOL composition transitioning to an oblate vesicle with complete bilayer formation within 300 ns. B) 68%POPC /29% CHOL/3% DLPC composition, where DLPC was placed in the open edges resulting in a delayed bilayer formation occurring at 400 ns. Cutouts reveal the topological transition from a single continuous monolayer to two separate lipid bilayer leaflets in each oblate vesicle.

edge-specific lipid placement can be applied.

Another application of open-edge geometries results from the periodic boundary conditions used in MD simulations. Membranes that form continuous structures in periodic boundary conditions appear as discontinuous, open-edged structures in Euclidean space. Since 3D modeling software typically operates in Euclidean space, meshes that should be continuous when simulated with periodic boundaries are frequently exported as open structures. Our new implementation addresses this limitation by properly handling these open-edged geometries within the periodic simulation environment. Lipid cubic phases are examples of such systems, where a membrane lying on a triply periodic minimal surface divides space into two bicontinuous water channels [47, 48]. Self-assembly MD simulations of single-unit cells of lipid cubic phases have provided insights into their structure and the dynamics of their

components [49–51]. However, building larger systems composed of multiple-unit cells has remained challenging [52, 53].

The triangulated mesh representing the diamond cubic phase was generated using the analytical solution of the Schwarz diamond minimal surface [54] employing the code supplied by Brasnett et al. [55]. Using monoolein at a water/lipid ratio of 0.29 w/w and 300 K, we constructed two systems, a single unit cell ($\pm 10\ nm^3$) and a $2 \times 2 \times 2$ periodic array of unit cells (visualized in Figure 7). Each system was simulated for 2 μs, throughout which the bilayer maintains close alignment with the mathematical minimal surface (Supplementary Figure SF2). The connectivity of the water channels was also analyzed using containment analysis, which showed that the non-intersecting water channels were also preserved (Supplementary Figure SF3) [56]. This model shows that TS2CG 2.0 enables the modeling of increasingly complex membranes, opening new avenues for studying systems from protein crystallization matrices to organelle membrane networks, where cubic phases play crucial biological roles [57–62].

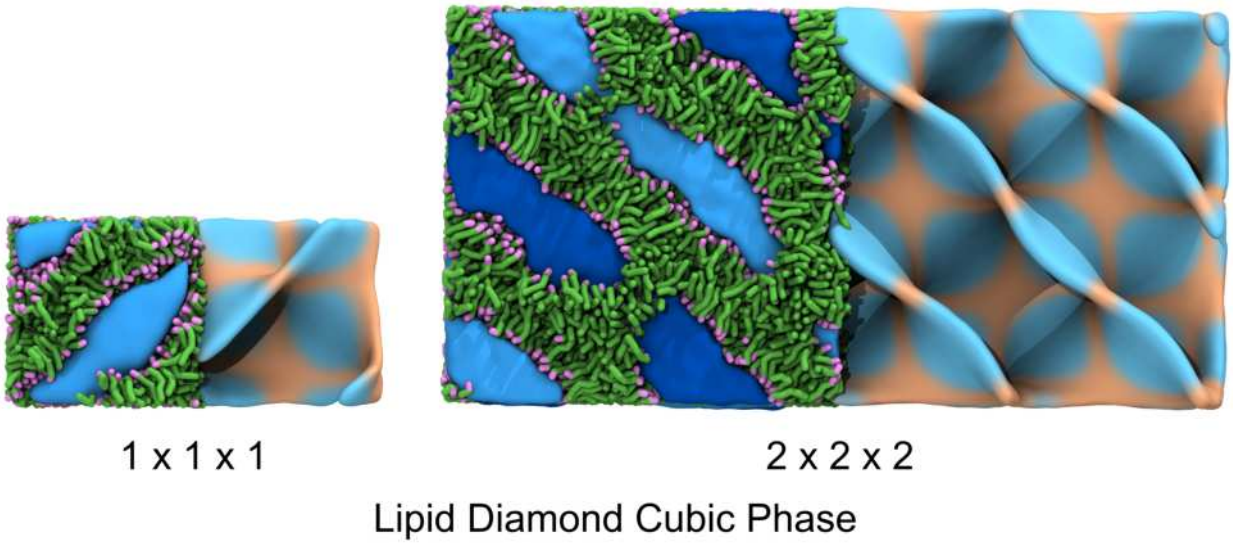

Figure 7: Monoolein diamonds Snapshots of a monoolein diamond cubic phase at two different lattice sizes ($1 \times 1 \times 1$ and $2 \times 2 \times 2$). The left panels show the molecular representation with lipid tails in green, headgroups in purple, and the water channels displayed in hues of blue. The right panels display the fitted surface of the cubic phase lipids colored according to their Gaussian curvature.

Martini Globe

In the last case study, the precision of the lipid placement is shown on an artificial case, in which the continents of our planet are mapped onto a membrane vesicle. In the final case study, although it lacks biological relevance, the precision of lipid placement is exemplified using an artificial example where Earth's continents are mapped onto a membrane vesicle. We used PLM to create a discrete point distribution from a spherical surface mesh. Each point's coordinates are converted to longitude and latitude, which are then mapped to continental boundaries using public geographic data obtained from US Geological Survey [63] and read by the regionmask python package [36, 64]. Points are assigned to a continent, i.e. lipid domain, based on their geographic coordinates, with oceans designated as domain 0. To distinguish the continental regions, we assign different lipid types to each domain. To demonstrate the fine-grained control of our method, we chose the system size so that Denmark, a relatively small country, can be represented by its own lipid domain. In the final model constructed, Denmark proper is defined by exactly three lipids. The Kingdom of Denmark is then assigned its own domain.

After the globe was created using TS2CG 2.0, it was equilibrated using dry martini [65]. The resulting model is shown in Figure 8. The details for the simulations can be found in the supplementary methods (S4 Methods). An artistic movie representation of the globe can be found in the supplementary material (S5).

Discussion and Conclusion

TS2CG, version 2.0, introduces significant extensions to its predecessor, emerging as a versatile coarse-grained membrane builder. This new version represents a major advance in membrane system preparation for molecular dynamics simulations. During the development process, alpha-versions of TS2CG 2.0 were already successfully used in other projects to simulate lipid nanoparticles [66, 67].

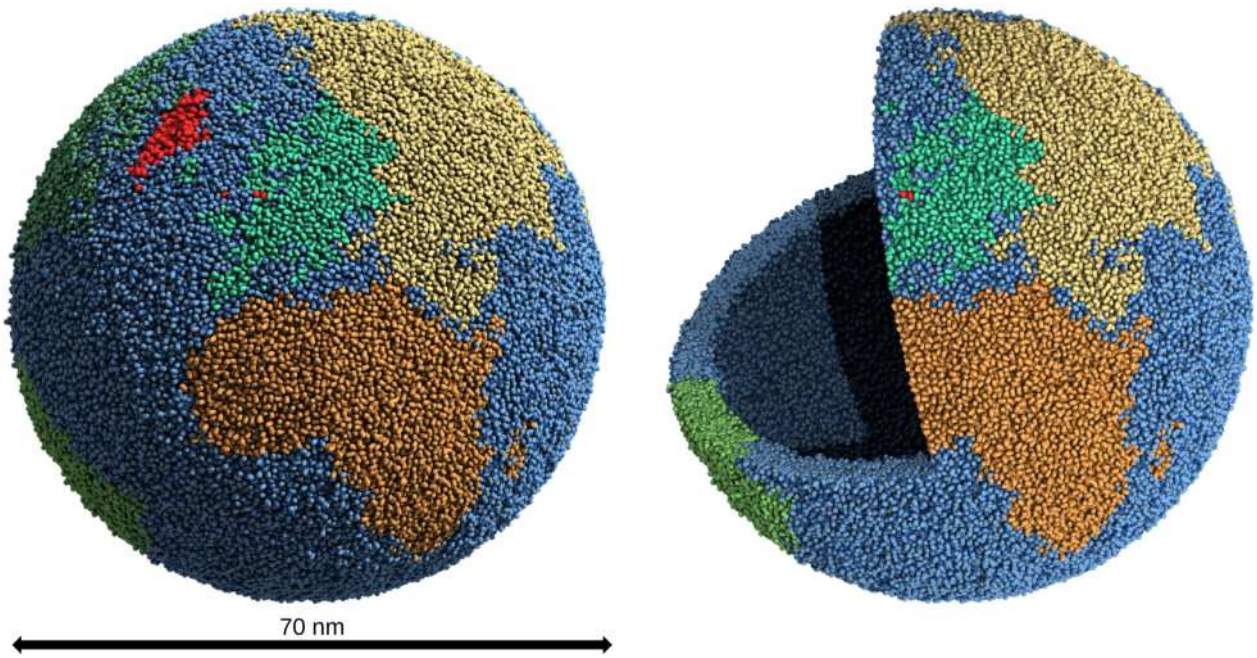

Figure 8: The Martini Globe The lipids of the globe colored by their continent based on the tectonic plates. The Kingdom of Denmark is colored dark red. Panel A shows the whole surface. Panel B shows a cutout of the globe visualizing the inner and outer monolayer and the lipid tails in between.

TS2CG 2.0 now supports open-edge geometries, removing the previous limitation to closed surfaces. This advancement enables easier compatibility with mesh generation tools like Blender, GAMer, and Geogram [33, 35, 68], expanding the range of feasible membrane architectures. We demonstrated the software's versatility through applications ranging from complex biological structures like mitochondrial membranes to specialized geometries such as lipid diamond cubic phases and a Möbius strip.

For simpler membrane architectures, TS2CG 2.0 introduces analytical membrane generation, enabling easy construction for a set of simple membrane shapes without requiring input meshes. These analytically generated membranes can be coupled with position restraints that preserve their precise geometry throughout MD simulation, which are referred to as wall beads (visualized in Figure 3C). This membrane scaffolding approach enables the systematic investigation of curvature-dependent lipid and protein sorting, generating new insights that inform the construction of more complex membrane architectures.

At its core, TS2CG 2.0 enables precise control over membrane lateral organization through both direct manipulation and curvature-based probabilistic approaches. The direct control over lipid organization is shown through the construction of a model vesicle that precisely maps different lipid types to Earth's continents, achieving a fine enough resolution to define Denmark proper with just three lipids. While this globe model serves as a technical demonstration, the same methodology enables the construction of biologically relevant membrane organizations, particularly generating phase-separated domains with geometric control. Note that the CG membrane models built by TS2CG can subsequently serve as templates for all-atom simulations through established backmapping protocols [69–71], also providing a robust pathway for generating complex membrane architectures at atomic resolution.

The biological relevance of these capabilities is exemplified in our mitochondrial membrane models. For the large-scale mitochondrial membrane, we showcase the curvature-based lipid distributions, while in the smaller cristae model, both membrane proteins and lipids are positioned according to their specific curvature preferences. This geometry-aware approach to membrane building enables the generation of closer-to-equilibrium starting configurations, increasing the stability of large-scale systems while reducing equilibration times in molecular dynamics simulations.

While TS2CG 2.0 provides control over membrane construction, its capabilities are ultimately limited by available molecular information. Our building protocol requires specific inputs about curvature preferences and molecular distributions, which are often only partially available from experiments. However, this limitation highlights an opportunity in integrative modeling. By combining TS2CG with experimental data from multiple sources - global proteomics, lipidomics, and structural studies - we can explore molecular-scale membrane organization that satisfies these experimental constraints. This approach provides insights into membrane architecture beyond the resolution of current experimental techniques. Future developments will likely incorporate machine learning approaches to predict membrane organization, trained

on existing simulations of simpler systems, further bridging the gap between global cellular measurements and molecular-scale architecture. When combined with molecular dynamics simulations, this integration would effectively create a true computational microscope for investigating membrane structure and dynamics in situ [8].

Performance is always a key concern for the computational modeling community. Here, we brought the performance of TS2CG 2.0 to light by constructing the entire inner mitochondrial membrane using curvature-biased lipid placement. Despite the current version of TS2CG being programmed to utilize only a single thread, it can still generate complex membranes within a feasible time frame. For instance, the mitochondrial membrane was created within 10 minutes. However, several features of the PLM and PCG routines could be optimized for multi-threading, enabling even larger membrane-building projects in the future. Implementing these improvements will require a refactoring of the C++ code base, which is already planned in our continuous development pipeline.

In conclusion, TS2CG 2.0 provides a user-friendly workflow that allows users to generate membranes of any composition to initialize complex coarse-grained membrane simulations. In short, TS2CG 2.0 makes constructing intricate membranes possible while also simplifying the creation of commonly used flat membrane simulations.

Acknowledgements

This research is supported by the Novo Nordisk Foundation (grant No. NNF18SA0035142 and NNF22OC0079182, NNF20OC0063808). W.P. is also supported by Independent Research Fund Denmark (grant No. 10.46540/2064-00032B) and acknowledges the funding from Marie Skodowska-Curie Fellowship (grant No. 101104867). S.M. and C.M.B. acknowledge funding from the ERC with the Advanced grant 101053661 ("COMP-O-CELL"). S.M. and J.A.S acknowledge funding from NWO through the NWA grant "The limits to growth: The challenge to dissipate energy".

Supporting Information

- Main supplementary document: Includes an overview of all available supplements (S1 - S6) and directly includes the methods and supplementary case study information.
- S1: A full-length pdf version of the TS2CG 2.0 tutorials
- S2: The general manual for TS2CG 2.0
- S3: The documentation for the newly developed python functions
- S5: An outreach video of the martini globe simulation (played in reverse)

References

[1] Paulo C T Souza, Riccardo Alessandri, Jonathan Barnoud, Sebastian Thallmair, Ignacio Faustino, Fabian Grünewald, Ilias Patmanidis, Haleh Abdizadeh, Bart M H Bruininks, Tsjerk A Wassenaar, Peter C Kroon, Josef Melcr, Vincent Nieto, Valentina Corradi, Hanif M Khan, Jan Domański, Matti Javanainen, Hector Martinez-Seara, Nathalie Reuter, Robert B Best, Ilpo Vattulainen, Luca Monticelli, Xavier Periole, D Peter Tieleman, Alex H de Vries, and Siewert J Marrink. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature Methods*, 18(4):382–388, March 2021.

[2] Florencia Klein, Martín Soñora, Lucianna Helene Santos, Ezequiel Nazareno Frigini, Andrés Ballesteros-Casallas, Matías Rodrigo Machado, and Sergio Pantano. The SIRAH force field: A suite for simulations of complex biological systems at the coarse-grained and multiscale level. *Journal of Structural Biology*, 215(3):107985, September 2023.

[3] Premila P Samuel Russell, Sepehr Alaeen, and Taras V Pogorelov. In-Cell dynamics: The next focus of All-Atom simulations. *The Journal of Physical Chemistry B*, 127, October 2023.

[4] Giulio Tesei, Anna Ida Trolle, Nicolas Jonsson, Johannes Betz, Frederik E Knudsen, Francesco Pesce, Kristoffer E Johansson, and Kresten Lindorff-Larsen. Conformational ensembles of the human intrinsically disordered proteome. *Nature*, 626(8000):897–904, February 2024.

[5] James C Phillips, David J Hardy, Julio D C Maia, John E Stone, João V Ribeiro, Rafael C Bernardi, Ronak Buch, Giacomo Fiorin, Jérôme Hénin, Wei Jiang, Ryan McGreevy, Marcelo C R Melo, Brian K Radak, Robert D Skeel, Abhishek Singharoy, Yi Wang, Benoît Roux, Aleksei Aksimentiev, Zaida Luthey-Schulten, Laxmikant V Kalé, Klaus Schulten, Christophe Chipot, and Emad Tajkhorshid. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J Chem Phys*, 153(4):044130, July 2020.

[6] Erik Lindahl, Berk Hess, and David van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *J. Mol. Model.*, 7(8):306–317, August 2001.

[7] Peter Eastman, Raimondas Galvelis, Raúl P Peláez, Charlles R A Abreu, Stephen E Farr, Emilio Gallicchio, Anton Gorenko, Michael M Henry, Frank Hu, Jing Huang, Andreas Krämer, Julien Michel, Joshua A Mitchell, Vijay S Pande, João Pglm Rodrigues, Jaime Rodriguez-Guerra, Andrew C Simmonett, Sukrit Singh, Jason Swails, Philip Turner, Yuanqing Wang, Ivy Zhang, John D Chodera, Gianni De Fabritiis, and Thomas E Markland. OpenMM 8: Molecular dynamics simulation with machine learning potentials. *J Phys Chem B*, 128(1):109–116, January 2024.

[8] Chelsea M. Brown and Siewert J. Marrink. Modeling membranes in situ. *Current Opinion in Structural Biology*, 87:102837, August 2024.

[9] Chelsea M Brown, Marieke S S Westendorp, Rubi Zarmiento-Garcia, Jan A Stevens, Bart M H Bruininks, Sarah L Rouse, Siewert J Marrink, and Tsjerk A Wassenaar. An integrative modelling approach to the mitochondrial cristae. *Communications Biology*, 8:972 2025.

[10] Abigail Dommer, Lorenzo Casalino, Fiona Kearns, Mia Rosenfeld, Nicholas Wauer, Surl-Hee Ahn, John Russo, Sofia Oliveira, Clare Morris, Anthony Bogetti, Anda Trifan, Alexander Brace, Terra Sztain, Austin Clyde, Heng Ma, Chakra Chennubhotla, Hyungro Lee, Matteo Turilli, Syma Khalid, Teresa Tamayo-Mendoza, Matthew Welborn, Anders Christensen, Daniel Ga Smith, Zhuoran Qiao, Sai K Sirumalla, Michael O'Connor, Frederick Manby, Anima Anandkumar, David Hardy, James Phillips, Abraham Stern, Josh Romero, David Clark, Mitchell Dorrell, Tom Maiden, Lei Huang, John McCalpin, Christopher Woods, Alan Gray, Matt Williams, Bryan Barker, Harinda Rajapaksha, Richard Pitts, Tom Gibbs, John Stone, Daniel M Zuckerman, Adrian J Mulholland, Thomas Miller, 3rd, Shantenu Jha, Arvind Ramanathan, Lillian Chong, and Rommie E Amaro. #COVIDisAirborne: AI-enabled multiscale computational microscopy of delta SARS-CoV-2 in a respiratory aerosol. *Int J High Perform Comput Appl*, 37(1):28–44, January 2023.

[11] Syma Khalid, Astrid F Brandner, Nikolai Juraschko, Kahlan E Newman, Conrado Pedebos, Dheeraj Prakaash, Iain P S Smith, Callum Waller, and Dhanushka Weerakoon. Computational microbiology of bacteria: Advancements in molecular dynamics simulations. *Structure*, 31(11):1320–1327, November 2023.

[12] Jan A Stevens, Fabian Grünewald, P A Marco van Tilburg, Melanie König, Benjamin R Gilbert, Troy A Brier, Zane R Thornburg, Zaida Luthey-Schulten, and Siewert J Marrink. Molecular dynamics simulation of an entire cell. *Front. Chem.*, 11:1106495, January 2023.

[13] Weria Pezeshkian, Melanie König, Tsjerk A Wassenaar, and Siewert J Marrink. Backmapping triangulated surfaces to coarse-grained membrane models. *Nature Communications*, 11(1):1–9, May 2020.

[14] Siewert J Marrink, Erik Lindahl, Olle Edholm, and Alan E Mark. Simulation of the

spontaneous aggregation of phospholipids into bilayers. *J Am Chem Soc*, 123(35):8638–8639, September 2001.

[15] Fabian Grünewald, Riccardo Alessandri, Peter C Kroon, Luca Monticelli, Paulo C T Souza, and Siewert J Marrink. Polyply; a python suite for facilitating simulations of macromolecules and nanomaterials. *Nat Commun*, 13(1):68, January 2022.

[16] Marieke S. S. Westendorp, Jan A. Stevens, Bart M.H. Bruininks, and Siewert-Jan Marrink. Bentopy: building molecular dynamics simulations with cellular complexity and scale., September 2024.

[17] Pin-Chia Hsu, Bart M H Bruininks, Damien Jefferies, Paulo Cesar Telles de Souza, Jumin Lee, Dhilon S Patel, Siewert J Marrink, Yifei Qi, Syma Khalid, and Wonpil Im. CHARMM-GUI martini maker for modeling and simulation of complex bacterial membranes with lipopolysaccharides. *J Comput Chem*, 38(27):2354–2363, October 2017.

[18] Martín Soñora, Leandro Martínez, Sergio Pantano, and Matías R Machado. Wrapping up viruses at multiscale resolution: Optimizing PACKMOL and SIRAH execution for simulating the zika virus. *J Chem Inf Model*, 61(1):408–422, January 2021.

[19] Harvey T McMahon and Emmanuel Boucrot. Membrane curvature at a glance. *J Cell Sci*, 128(6):1065–1070, March 2015.

[20] Weria Pezeshkian and Siewert J Marrink. Simulating realistic membrane shapes. *Curr Opin Cell Biol*, 71:103–111, August 2021.

[21] Markus Deserno. Biomembranes balance many types of leaflet asymmetries. *Curr Opin Struct Biol*, 87:102832, August 2024.

[22] Mikkel Dahl Andreasen, Paulo C T Souza, Birgit Schott, and Lorena Zuzic. Creating coarse-grained systems with COBY: Towards higher accuracy in membrane complexity.

bioRxiv, July 2024.

[23] Tsjerk A Wassenaar, Helgi I Ingólfsson, Rainer A Böckmann, D Peter Tieleman, and Siewert J Marrink. Computational lipidomics with insane: A versatile tool for generating custom membranes for molecular simulations. *J Chem Theory Comput*, 11(5):2144–2155, May 2015.

[24] Jacob D Durrant and Rommie E Amaro. LipidWrapper: an algorithm for generating large-scale membrane models of arbitrary geometry. *PLoS Comput Biol*, 10(7):e1003720, July 2014.

[25] Noah Trebesch and Emad Tajkhorshid. Enabling atomistic modeling and simulation of complex curved cellular membranes with xMAS builder. *bioRxiv*, January 2025.

[26] Kevin J Boyd and Eric R May. BUMPy: A Model-Independent tool for constructing lipid bilayers of varying curvature and composition. *J Chem Theory Comput*, 14(12):6642–6652, December 2018.

[27] Weria Pezeshkian, Melanie König, Siewert J Marrink, and John H Ipsen. A Multi-Scale approach to membrane remodeling processes. *Front Mol Biosci*, 6:59, July 2019.

[28] Weria Pezeshkian and John H Ipsen. Mesoscale simulation of biomembranes with FreeDTS. *Nature Communications*, 15(1):1–10, January 2024.

[29] M König, R de Vries, F Grünewald, S J Marrink, and W Pezeshkian. Curvature-induced lipid sorting beyond the critical packing parameter. *bioRxiv*, December 2023.

[30] Elio A Cino and D Peter Tieleman. Curvature-based sorting of eight lipid types in asymmetric buckled plasma membrane models. *Biophys J*, 121(11):2060–2068, June 2022.

[31] Julie Cornet, Nelly Coulonges, Weria Pezeshkian, Maël Penissat-Mahaut, Hermes Desgrez-Dautet, Siewert J Marrink, Nicolas Destainville, Matthieu Chavent, and Manoel Manghi.

There and back again: bridging meso- and nano-scales to understand lipid vesicle patterning. *Soft Matter*, 20(25):4998–5013, June 2024.

[32] A D MacKerell, D Bashford, M Bellott, R L Dunbrack, J D Evanseck, M J Field, S Fischer, J Gao, H Guo, S Ha, D Joseph-McCarthy, L Kuchnir, K Kuczera, F T Lau, C Mattos, S Michnick, T Ngo, D T Nguyen, B Prodhom, W E Reiher, B Roux, M Schlenkrich, J C Smith, R Stote, J Straub, M Watanabe, J Wiórkiewicz-Kuczera, D Yin, and M Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J Phys Chem B*, 102(18):3586–3616, April 1998.

[33] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[34] Mesh. https://github.com/BrunoLevy/geogram/wiki/Mesh. Accessed: 2025-2-7.

[35] Christopher T Lee, Justin G Laughlin, Nils Angliviel de La Beaumelle, Rommie E Amaro, J Andrew McCammon, Ravi Ramamoorthi, Michael Holst, and Padmini Rangamani. 3D mesh processing using GAMer 2 to enable reaction-diffusion simulations in realistic cellular geometries. *PLoS Comput Biol*, 16(4):e1007756, April 2020.

[36] Aric A. Hagberg and Daniel A. Schult and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15, 2008.

[37] Mark Abraham, Andrey Alekseenko, Brian Andrews, Vladimir Basov, Paul Bauer, Hugh Bird, Eliane Briand, Ania Brown, Mahesh Doijade, Giacomo Fiorin, Stefan Fleischmann, Sergey Gorelov, Gilles Gouaillardet, Alan Gray, M. Eric Irrgang, Farzaneh Jalalypour, Petter Johansson, Carsten Kutzner, Grzegorz Łazarski, Justin A. Lemkul, Magnus Lundborg, Pascal Merz, Vedran Miletić, Dmitry Morozov, Lukas Müllender, Julien Nabet, Szilárd Páll, Andrea Pasquadibisceglie, Michele Pellegrino, Nicola Piasentin,

Daniele Rapetti, Muhammad Umair Sadiq, Hubert Santuz, Roland Schulz, Michael Shirts, Tatiana Shugaeva, Alexey Shvetsov, Philip Turner, Alessandra Villa, Sebastian Wingbermühle, Berk Hess, and Erik Lindahl. Gromacs 2025.1 manual, March 2025.

[38] Fabian Schuhmann, Leonie Ryvkin, James D McLaren, Luca Gerhards, and Ilia A Solov'yov. Across atoms to crossing continents: Application of similarity measures to biological location data. *PLoS One*, 18(5):e0284736, May 2023.

[39] Kate McArthur and Michael T Ryan. Resolving mitochondrial cristae: introducing a new model into the fold. *EMBO J*, 39(14):e105714, July 2020.

[40] Guadalupe C Garcia, Thomas M Bartol, Sébastien Phan, Eric A Bushong, Guy Perkins, Terrence J Sejnowski, Mark H Ellisman, and Alexander Skupin. Mitochondrial morphology provides a mechanism for energy buffering at synapses. *Sci Rep*, 9(1):18306, December 2019.

[41] Lea Dietrich, Ahmed-Noor A Agip, Christina Kunz, Andre Schwarz, and Werner Kühlbrandt. In situ structure and rotary states of mitochondrial ATP synthase in whole cells. *Science*, 385(6713):1086–1090, September 2024.

[42] Michael Schlame. Protein crowding in the inner mitochondrial membrane. *Biochim Biophys Acta Bioenerg*, 1862(1):148305, January 2021.

[43] Werner Kühlbrandt. Structure and function of mitochondrial membrane protein complexes. *BMC Biol*, 13:89, October 2015.

[44] Matthew Zorkau, Christin A Albus, Rolando Berlinguer-Palmini, Zofia M A Chrzanowska-Lightowlers, and Robert N Lightowlers. High-resolution imaging reveals compartmentalization of mitochondrial protein synthesis in cultured human cells. *Proc Natl Acad Sci U S A*, 118(6), February 2021.

[45] Matthias F Bauer, Sabine Hofmann, Walter Neupert, and Michael Brunner. Protein translocation into mitochondria: the role of TIM complexes. *Trends Cell Biol*, 10(1):25–31, January 2000.

[46] Marcel Morgenstern, Christian D Peikert, Philipp Lübbert, Ida Suppanz, Cinzia Klemm, Oliver Alka, Conny Steiert, Nataliia Naumenko, Alexander Schendzielorz, Laura Melchionda, Wignand W D Mühlhäuser, Bettina Knapp, Jakob D Busch, Sebastian B Stiller, Stefan Dannenmaier, Caroline Lindau, Mariya Licheva, Christopher Eickhorst, Riccardo Galbusera, Ralf M Zerbes, Michael T Ryan, Claudine Kraft, Vera Kozjak-Pavlovic, Friedel Drepper, Sven Dennerlein, Silke Oeljeklaus, Nikolaus Pfanner, Nils Wiedemann, and Bettina Warscheid. Quantitative high-confidence human mitochondrial proteome and its dynamics in cellular context. *Cell Metab*, 33(12):2464–2483.e18, December 2021.

[47] S T Hyde and Sten Andersson. A cubic structure consisting of a lipid bilayer forming an infinite periodic minimum surface of the gyroid type in the glycerolmonooleat-water system. *Z. Kristallogr. Cryst. Mater.*, 168(1-4):213–220, November 1984.

[48] Charlotte E Conn and Calum J Drummond. Nanostructured bicontinuous cubic lipid self-assembly materials as matrices for protein encapsulation. *Soft Matter*, 9(13):3449–3464, March 2013.

[49] S J Marrink and D P Tieleman. Molecular dynamics simulation of a lipid diamond cubic phase. *J. Am. Chem. Soc.*, 123(49):12383–12391, December 2001.

[50] Niklaus Johner, Sayan Mondal, Giulia Morra, Martin Caffrey, Harel Weinstein, and George Khelashvili. Protein and lipid interactions driving molecular mechanisms of in meso crystallization. *J Am Chem Soc*, 136(8):3271–3284, February 2014.

[51] George Khelashvili, Pedro Blecua Carrillo Albornoz, Niklaus Johner, Sayan Mondal, Martin Caffrey, and Harel Weinstein. Why GPCRs behave differently in cubic and

lamellar lipidic mesophases. *J Am Chem Soc*, 134(38):15858–15868, September 2012.

[52] Kasper B Pedersen, Helgi I Ingólfsson, Daniel P Ramirez-Echemendia, Luís Borges-Araújo, Mikkel D Andreasen, Charly Empereur-mot, Josef Melcr, Tugba N Ozturk, Drew W F Bennett, Lisbeth R Kjølbye, Christopher Brasnett, Valentina Corradi, Hanif M Khan, Elio A Cino, Jackson Crowley, Hyuntae Kim, Balázs Fábián, Ana C Borges-Araújo, Giovanni M Pavan, Fabio Lolicato, Tsjerk A Wassenaar, Manuel N Melo, Sebastian Thallmair, Timothy S Carpenter, Luca Monticelli, D Peter Tieleman, Birgit Schiøtt, Paulo C T Souza, and Siewert J Marrink. The martini 3 lipidome: Expanded and refined parameters improve lipid phase behavior. *ChemRxiv*, December 2024.

[53] Marc Fuhrmans, Volker Knecht, and Siewert J Marrink. A single bicontinuous cubic phase induced by fusion peptides. *J Am Chem Soc*, 131(26):9166–9167, July 2009.

[54] U S Schwarz and G Gompper. Systematic approach to bicontinuous cubic phases in ternary amphiphilic systems. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, 59(5 Pt B):5528–5541, May 1999.

[55] cubic-phase-fitter. https://github.com/csbrasnett/cubic-phase-fitter. Accessed: 2025-04-15.

[56] GitHub - BartBruininks/mdvcontainment: Robust characterization of inside and outside in periodic spaces. https://github.com/BartBruininks/mdvcontainment. Accessed: 2025-2-6.

[57] Tomas Landh. From entangled membranes to eclectic morphologies: cubic membranes as subcellular space organizers. *FEBS Lett*, 369(1):13–17, August 1995.

[58] G Rummel, A Hardmeyer, C Widmer, M L Chiu, P Nollert, K P Locher, I Pedruzzi, I, E M Landau, and J P Rosenbusch. Lipidic cubic phases: New matrices for the

Three-Dimensional crystallization of membrane proteins. *J Struct Biol*, 121(2):82–91, 1998.

[59] Raffaele Mezzenga, John M Seddon, Calum J Drummond, Ben J Boyd, Gerd E Schröder-Turk, and Laurent Sagalowicz. Nature-Inspired design and application of lipidic lyotropic liquid crystals. *Adv Mater*, 31(35):e1900818, August 2019.

[60] Alexandru Zabara, Josephine Tse Yin Chong, Isabelle Martiel, Laura Stark, Brett A Cromer, Chiara Speziale, Calum John Drummond, and Raffaele Mezzenga. Design of ultra-swollen lipidic mesophases for the crystallization of membrane proteins with large extracellular domains. *Nat Commun*, 9(1):544, February 2018.

[61] Hanna M G Barriga, Margaret N Holme, and Molly M Stevens. Cubosomes: The next generation of smart lipid nanoparticles? *Angew Chem Int Ed Engl*, 58(10):2958–2978, March 2019.

[62] Annela M. Seddon. Chapter six - recent developments in the production, analysis, and applications of cubic phases formed by lipids. volume 18 of *Advances in Planar Lipid Bilayers and Liposomes*, pages 147–180. Academic Press, 2013.

[63] Environmental Systems Research Institute, Inc. (ESRI). World continents. Vector digital data, ESRI Data & Maps 2002 (CD 1) \world, 2002. Edition: 2002.

[64] GitHub - regionmask/regionmask: create masks of geospatial regions for arbitrary grids. https://github.com/regionmask/regionmask. Accessed: 2025-1-18.

[65] Clément Arnarez, Jaakko J Uusitalo, Marcelo F Masman, Helgi I Ingólfsson, Djurre H de Jong, Manuel N Melo, Xavier Periole, Alex H de Vries, and Siewert J Marrink. Dry martini, a Coarse-Grained force field for lipid membrane simulations with implicit solvent. *J. Chem. Theory Comput.*, 11(1):260–275, 2014.

[66] Markéta Paloncýová, Mariana Valério, Ricardo Nascimento Dos Santos, Petra Kührová, Martin Šrejber, Petra Čechová, Dimitar A Dobchev, Akshay Balsubramani, Pavel Banáš, Vikram Agarwal, Paulo C T Souza, and Michal Otyepka. Computational methods for modeling Lipid-Mediated active pharmaceutical ingredient delivery. *Mol Pharm*, 22(3):1110–1141, March 2025.

[67] Lisbeth Ravnkilde Kjølbye, Mariana Valério, Markéta Paloncýová, Luís Borges-Araújo, Roberto Pestana-Nobles, Fabian Grünewald, Bart M H Bruininks, Rocío Araya-Osorio, Martin Šrejber, Raul Mera-Adasme, Luca Monticell, Siewert J Marrink, Michal Otyepka, Sangwook Wu, and Paulo C T Souza. Martini 3 building blocks for lipid nanoparticle design. *ChemRxiv*, 2025. Preprint, Version 2, posted January 2, 2025.

[68] GitHub - BrunoLevy/geogram: a programming library with geometric algorithms. https://github.com/BrunoLevy/geogram. Accessed: 2025-2-13.

[69] Tsjerk A Wassenaar, Kristyna Pluhackova, Rainer A Böckmann, Siewert J Marrink, and D Peter Tieleman. Going backward: A flexible geometric approach to reverse transformation from coarse grained to atomistic models. *J Chem Theory Comput*, 10(2):676–690, February 2014.

[70] Owen N Vickery and Phillip J Stansfeld. CG2AT2: an enhanced Fragment-Based approach for serial multi-scale molecular dynamics simulations. *J Chem Theory Comput*, 17(10):6472–6482, October 2021.

[71] Siyoung Kim. All-Atom membrane builder via multiscale simulation. *J Chem Inf Model*, 64(18):7077–7085, September 2024.

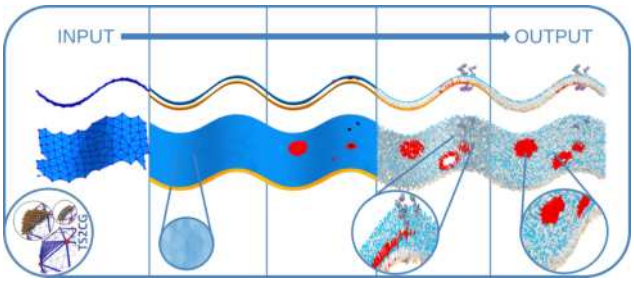

Figure 9: For Table of Contents Only