GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Biomolecular space exploration

Phase space and parameter space sampling in realistic protein-lipid environments

Dissertation for the award of the degree

Doctor rerum naturalium

of the Georg-August Universität Göttingen

within the doctoral program Physics

of the Georg-August-University School of Science (GAUSS)

submitted by

Kai Steffen Stroh

born in Gießen

Göttingen 2023

Thesis Advisory Committee

Prof. Dr. Herre Jelger Risselada
Institut für Theoretische Physik, Georg-August-Universität Göttingen
Fakultät Physik, Technische Universität Dortmund

Prof. Dr. Stefan Klumpp
Institut für Dynamik komplexer Systeme, Georg-August-Universität Göttingen

Prof. Dr. Matthias Krüger
Institut für Theoretische Physik, Georg-August-Universität Göttingen

Members of the Examination Board

Prof. Dr. Herre Jelger Risselada (1st Referee)
Institut für Theoretische Physik, Georg-August-Universität Göttingen
Fakultät Physik, Technische Universität Dortmund

Prof. Dr. Stefan Klumpp (2nd Referee)
Institut für Dynamik komplexer Systeme, Georg-August-Universität Göttingen

Further members of the Examination Board

Prof. Dr. Matthias Krüger
Institut für Theoretische Physik, Georg-August-Universität Göttingen

Prof. Dr. Michael Meinecke
Heidelberg University Biochemistry Center

Dr. Luca Monticelli
CNRS, INSERM, University of Lyon (France)

Prof. Dr. Marcus Müller
Institut für Theoretische Physik, Georg-August-Universität Göttingen

Date of oral examination: 23.01.2023

List of Papers

Paper 1

Kai Steffen Stroh and Herre Jelger Risselada
"Quantifying membrane curvature sensing of peripheral proteins by simulated buckling and umbrella sampling". In: *Journal of Chemical Theory and Computation*. Vol. 17, no. 8 (2021), pp. 5276–5286.
DOI: 10.1021/acs.jctc.1c00021.

Author contribution: KSS developed and implemented the method, performed and analyzed all MD simulations. KSS and HJR wrote the manuscript.

Paper 2

Fereshteh Sadeqi, Kai Steffen Stroh, Marian Vache, Dietmar Riedel, Andreas Janshoff, Herre Jelger Risselada and Michael Meinecke
"Membrane interactions of mitochondrial lipid transfer proteins".
In: *bioRxiv* (2022),
DOI: 10.1101/2022.04.05.487160.

Author contribution: KSS performed and analyzed all MD simulations, and wrote the corresponding section of the manuscript.

Paper 3

Niek van Hilten, Kai Steffen Stroh and Herre Jelger Risselada
"Efficient quantification of lipid packing defect sensing by amphipathic peptides: Comparing Martini 2 and 3 with CHARMM36". In: *Journal of Chemical Theory and Computation*. Vol. 18, no. 7 (2022), pp. 4503–4514.
DOI: 10.1021/acs.jctc.2c00222

Author contribution: KSS performed and analyzed all MD simulations related to the buckled membrane. Manuscript writing and editing (supporting).

Paper 4

Kai Steffen Stroh, Paulo C. T. Souza, Luca Monticelli and Herre Jelger Risselada "CGCompiler: Automated coarse-grained molecule parameterization via noise-resistant mixed-variable optimization".
In: *Journal of Chemical Theory and Computation*. Vol. 19, no. 22 (2023), pp. 8384-8400.
DOI: 10.1021/acs.jctc.3c00637.

Author contribution: KSS developed and implemented the method, performed and analyzed all MD simulations. KSS and HJR wrote the manuscript.

Paper 5

Niek van Hilten, Kai Steffen Stroh and Herre Jelger Risselada
"Membrane thinning induces sorting of lipids and the Amphipathic Lipid Packing Sensor (ALPS) protein motif". In: *Frontiers in Physiology*. Vol. 11, (2020), DOI: 10.3389/fphys.2020.00250

Author contribution: KSS: Theoretical aspects of free energy calculations; Manuscript writing and editing (supporting).

Paper 6

Denise Mehner-Breitfeld, Michael T. Ringel, Daniel Alexander Tichy, Laura J. Endter, Kai Steffen Stroh, Heinrich Lünsdorf, Herre Jelger Risselada and Thomas Brüser "TatA and TatB generate a hydrophobic mismatch important for the function and assembly of the Tat translocon in Escherichia coli". In: *Journal of Biological Chemistry*. Vol. 298, no. 9 (2022), pp. 102236. DOI: 10.1016/j.jbc.2022.102236.

Author contribution: KSS: co-supervision of DAT (BSc student); validation, formal analysis and visualization of MD simulations; writing–review and editing (supporting);

Contents

Chapter 1

General introduction

All living matter is based on four classes of small molecules: sugars, amino acids, nucleotides, and fatty acids. These elementary building blocks are combined with molecules of the same class or from the other classes to create larger entities: poly-saccharides, proteins, nucleic acids, and lipids [1]. Two of those building blocks of life, namely lipids and proteins, are the focus of this thesis.

1.1 The interplay of lipids and proteins

Due to their amphipathic nature, lipids in aqueous solution aggregate into supramolecular assemblies such as bilayers or micelles. Lipid bilayers confine and compartmentalize all living cells [1], but the enormous chemical diversity of lipids and compositional diversity of membranes is not required for barrier functions alone [2]. In fact, lipid composition affects membrane properties and specifically adapted compositions allow biomembranes to fulfill their many functions [3] and an altered composition is associated with disease [4].

Lipids consist of a polar headgroup and apolar hydrocarbon chains. In mammalian membranes, there are three major classes of lipids, glycerophospholipids (GPL), sphingolipids (SL) and sterols [3]. With different headgroups, varying length, degree of saturation, and hydroxylation of the hydrocarbon chains, thousands of different combinations are possible [3]. The structural differences of individual lipids and specific lipid compositions affect the physical properties of membranes through lipid-lipid and lipid-protein interactions [2]. Figure 1.1 shows two examples that are relevant in later chapters of this thesis. In the absence of other curvature regulating factors, such as lateral pressure or curvature-inducing proteins, membrane curvature is determined by the overall lipid shape, or more specifically, the volume ratio of headgroup and hydrocarbon chains [2]. Membrane fluidity and phase behavior is largely affected by the acyl chains' length and degree of saturation [5, 6]. Saturated acyl chains are easily ordered, resulting in tightly packed gel phases. The effect of cholesterol is complex, it has an ordering effect, but hinders the tight packing required for the formation of the gel phase at physiological temperatures. The combination of saturated acyl chains and cholesterol is therefore a so-called liquid ordered phase. The kinks in unsatured acyl chains cause disorder and concomitantly fluidize the membrane.

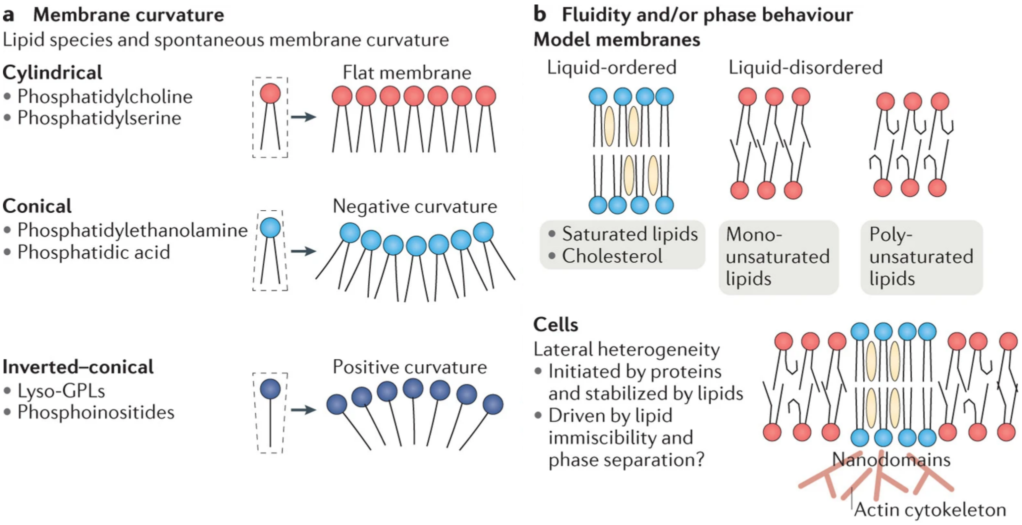
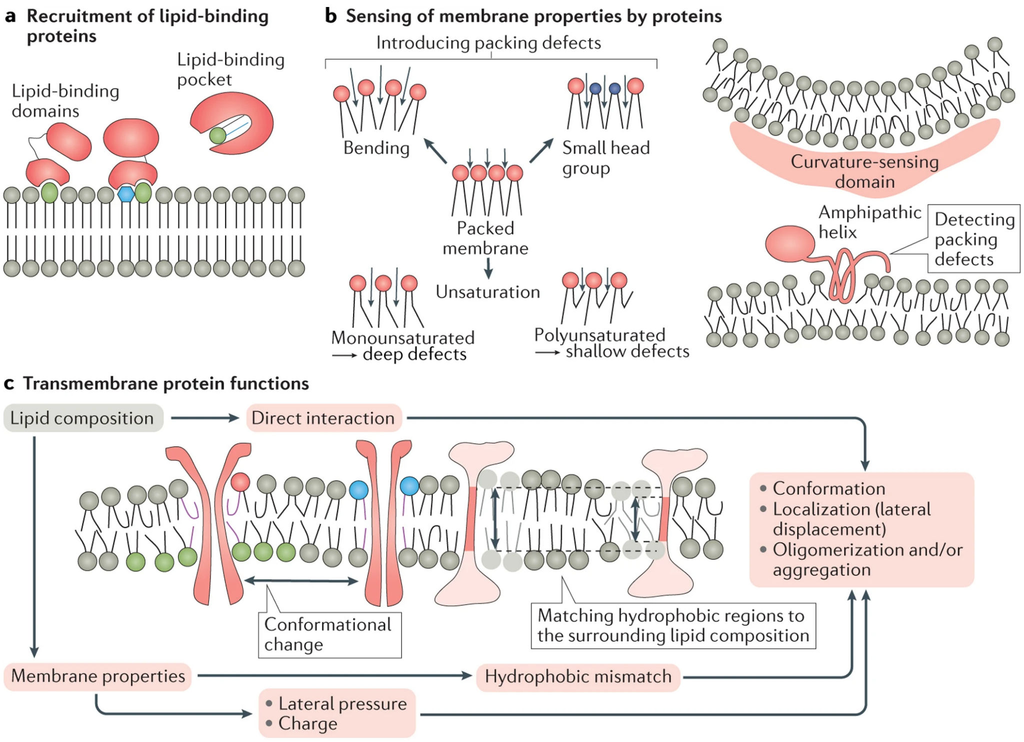


Figure 1.1: Lipids regulate biological processes through membrane properties. **a**) The size balance of headgroup and hydrophobic tails determines spontaneous curvature of membranes. Membrane curvature regulates protein sorting and might additionally be important for membrane remodeling. **b**) Degree of unsaturation affects membrane fluidity. Saturated tails and cholesterol form liquid-ordered phases. Unsaturation in acyl chains fluidizes the membrane. Adapted by permission from [2]. Copyright 2018, Springer Nature

Beyond curvature and fluidity, lipid properties affect membrane thickness, lateral pressure, surface charge and interdigitation [3]. Lipid composition also controls the energetics of stalk formation, the first step in membrane fusion [7].

Often, the regulating influence of lipids on biological processes is mediated by proteins [2]. Lipids can recruit proteins with specific lipid-binding domains (cf. Figure 1.2a), either by targeting single lipids, or cooperative binding to multiple lipids. Through membrane curvature and the related lipid packing defects, i.e., hydrophobic regions exposed to the environment, information about membrane properties is passed on to proteins (cf. Figure 1.2b). While the sensing ability of banana-shaped BAR domains [8] is reliant on matching geometrical shapes, packing defects allow sensing of membrane properties such as curvature, but also composition and tension, by smaller sensing domains. A substantial part of this thesis (Chapter 2) and the related publications (Papers 2,3,5) [9–11] are devoted to this topic. The role of hydrophobic mismatch in protein function regulation is explored in Paper 6 ([12]).

These examples show that lipid diversity profoundly affects biological processes. Therefore, a broad range of computational studies, where membrane-related


a Recruitment of lipid-binding proteins
Lipid-binding domains
Lipid-binding pocket
b Sensing of membrane properties by proteins
Introducing packing defects
Bending
Small head group
Packed membrane
Unsaturation
Monounsaturated → deep defects
Polyunsaturated → shallow defects
Curvature-sensing domain
Amphipathic helix
Detecting packing defects
c Transmembrane protein functions
Lipid composition
Direct interaction
Conformational change
Matching hydrophobic regions to the surrounding lipid composition
• Conformation
• Localization (lateral displacement)
• Oligomerization and/or aggregation
Membrane properties
Hydrophobic mismatch
• Lateral pressure
• Charge

Nature Reviews | Molecular Cell Biology

Figure 1.2: Lipids regulate protein-mediated biological processes. **a**) Specific lipid binding can recruit proteins. Additionally, lipids can be ligands for proteins, thereby regulating protein activity. **b**) Membrane curvature and lipid packing defects can be sensed by proteins. **c**) Lipid composition affects conformation, distribution and oligomerization of transmembrane proteins. The interaction can be direct by lipid-protein interaction, or indirect through membrane properties. Reprinted by permission from [2]. Copyright 2018, Springer Nature

effects play a role, rely on a large variety of lipid models to accurately recreate specific types of membranes. The endeavor for more realistic membrane models has received significant attention in recent years [13–17].

Lack of a Martini 3 sphingolipid model, a lipid class that constitutes approximately 30 mol% of the plasma membrane lipids [18], to study composition dependent protein binding to membranes, resulted in the other major subject of this thesis: automated parameterization of molecules for coarse-grained force fields.

1.2 Methods

One of the fundamental choices that has to be made in any kind of modeling of physical processes is the level of detail of the model. Evidently, all relevant degrees of freedom need to be retained, whereas the rest can and should be integrated out. Figure 1.3 contextualizes the accessible length and time scales of biophysical modeling methods. In the quantum resolution, processes such as the formation and breaking of covalent bonds can be modeled, but the quantum mechanical description is computationally so demanding, that simulation of lipid membranes is out of reach, even with approximations [19]. On the other end of the spectrum are continuum models, where rather large time and length scales are accessible, but all molecular detail is lost [19]. Rather than providing a detailed description of each of these methods, the choice of the here-employed coarse-grained description is motivated on the basis of some of the phenomena introduced in Section 1.1.

On the one hand, it is immediately clear that regulating effects of lipids that are mediated through a change in membrane properties cannot be studied in a quantum resolution, as the time and length scales involved are too large. On the other hand, the model resolution has to be fine enough, such that chemical specificity is maintained. This can be showcased on the basis of packing defect sensing. The sensing ability of amphipathic peptides relies on sticking individual amino acid side chains into these defects. Hence, at least a near to atomistic model is required to accurately resolve the details of this process. Due to the subtleness of packing defect or curvature sensing, i.e., small sorting forces, long sampling times are required. Therefore, usage of full atomistic resolutions is feasible regarding the length scales but often limiting in terms of time scales, for these kinds of processes. Mild coarse-graining approaches, where a few atoms are mapped into one supra-atom, can provide the necessary chemical specificity in combination with efficient sampling. Typically, a speed up of a factor of 100-1000 of coarse-grained molecular dynamics (CGMD) simulations compared to all-atom (AA) MD is achieved, based on the reduced number of particles, an increased time step, and a smoother energy landscape. The Martini force field (FF) [20, 21], that regularly utilizes a 4-1 mapping, but also 3-1 and 2-1 if a finer resolution is required, is one example of mild coarse-graining. In a recent comparison of the Martini FF and the atomistic CHARMM36 FF, we showed that packing defect sensing is in fact well reproduced in the somewhat coarser resolution [10].

While a broader introduction to coarse-graining in biomolecular simulations is given in Chapter 3, Section 1.2.1 aims to familiarize readers from outside this field with the philosophy of Martini's building block approach.

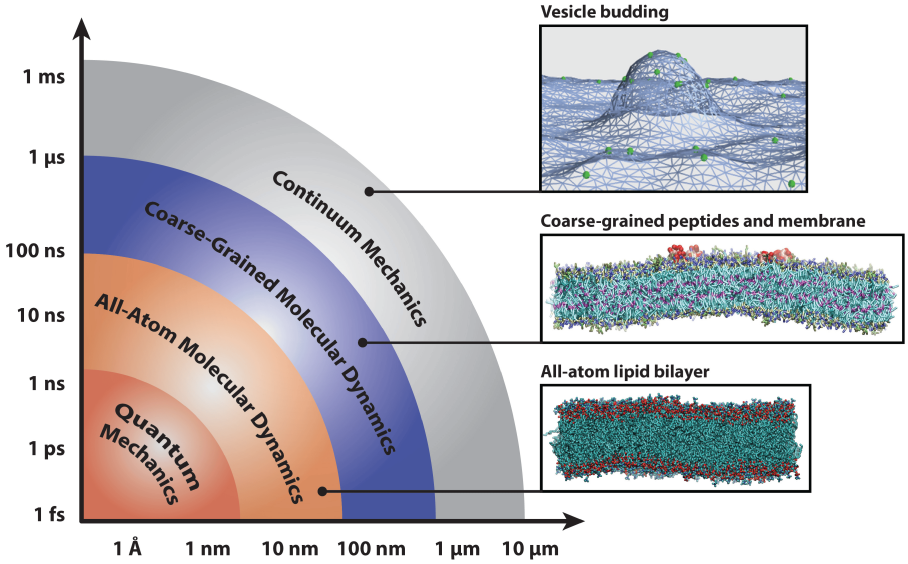


Figure 1.3: Different length and time scales in simulations regarding cellular processes. Reprinted from [22] (CC BY 4.0).

1.2.1 Molecular Lego: The building block approach in the Martini coarse-grained force field

The Martini coarse-grained force-field [20, 21] is a building block force-field, i.e., common chemical groups are parameterized as basic building blocks, which can be combined to build up a large variety of molecules. These basic building blocks of Martini, the beads, are parameterized top-down, with liquid-liquid partitioning and miscibility as main targets, while complete molecules are parameterized with a combination of top-down (experimental data) and bottom-up (atomistic simulation). Beyond the bead level, the building block approach also applies to larger molecular fragments, e.g., a hydrocarbon chain has the same mapping and interaction parameters, no matter if it is part of a glycerophospholipid, a sphingolipid, or not attached to a lipid at all. Conserving molecular compatibility, and hereby simplifying application to new problems, is the *raison d'être* of the building block approach.

Non-bonded interactions in the Martini FF are described by Lennard-Jones (LJ) and Coulomb potentials

$$V_{\text{non-bonded}}(r_1, ..., r_N) = \sum_{i,j} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] + \sum_{i,j} \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}}. \tag{1.1}$$

Having a fixed set of chemical bead types and sizes directly results in having a

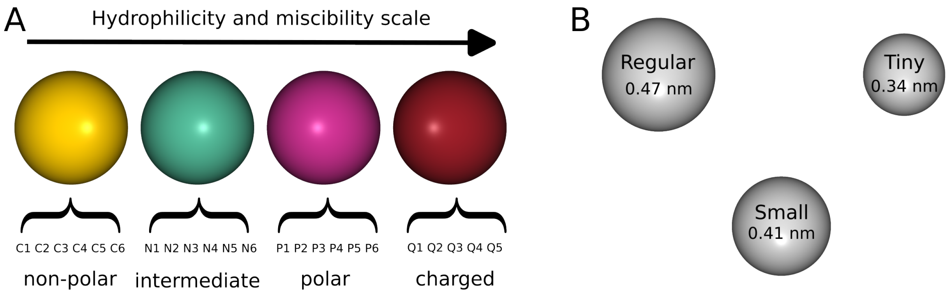


Figure 1.4: A: Trends of Martini 3 organic and ion bead types. B: Bead sizes. Recreated from [21]

fixed set of ϵ and σ values. Additionally, Martini 3 uses a combination rule only for bead sizes (σ_{ij} values). The interaction strength ϵ_{ij} is specifically defined for each pair of bead types. For super repulsive interactions, also σ_{ij} is increased.

The generic beads, used to represent all kinds of different chemical moieties, are arranged in chemical and size classes following trends in hydrophilicity and miscibility [21], cf. Figure 1.4, meaning polar and charged beads interact strongly with water and other polar or charged beads. Non-polar beads interact strongly with other non-polar beads. Cross-group interactions are weak. Correctly balancing the different interaction levels, is what actually makes the partitioning and miscibility, and also more complex phenomena, such as self-assembly of lipids into membranes, work. Particularly, since the entropic part of the hydrophobic effect is missing due to the coarse-grained description, it has to be recovered by modified enthalpic terms.

Construction of larger molecules is done by defining **bonded interactions** between beads.

$$\begin{aligned} V_{\text{bonded}} = \sum_{\text{bonds}} \frac{1}{2} k^{\text{bond}} (r - r_0)^2 + \sum_{\text{angles}} \frac{1}{2} k^{\text{angle}} (\cos\theta - \cos\theta_0)^2 \\ + \sum_{n,\text{dihedrals}} k^{\text{dihedral}} \left[1 + \cos(n\phi - \phi_s)\right] \end{aligned} \tag{1.2}$$

Martini 3 uses a center-of-geometry (COG) based mapping to define the position of beads. Importantly, parameterization of bonded and non-bonded parameters should be performed simultaneously since bonded and non-bonded interactions are not independent – they are directly influencing each other via the density of interactions [21, 23, 24]. This effect can be easily exemplified by varying the equilibrium bond length b_0 of molecules consisting of two C1 beads, which are solvated in single C1 beads. With an equilibrium bond length $b_0 = 0.47$ nm, which

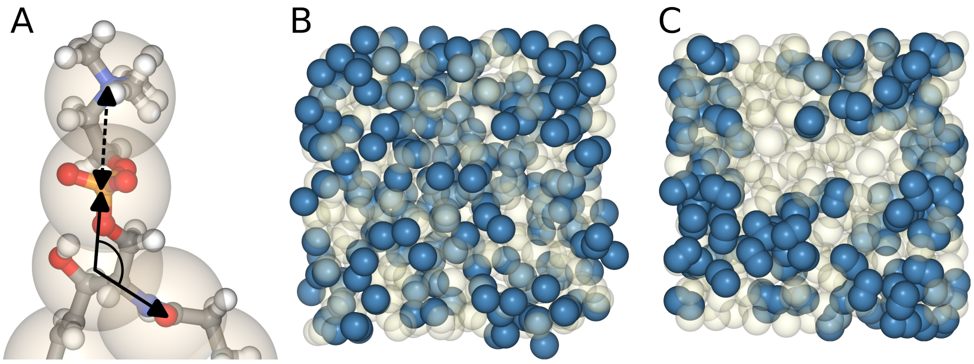


Figure 1.5: Martini 3 bonded interactions. **A** | Center of geometry mapping. **B-C** | C1-C1 molecules solvated in C1 beads, **B** C1-C1 equilibrium bond length $b_0 = 0.47\,\text{nm}$, **C** | C1-C1 equilibrium bond length $b_0 = 0.20\,\text{nm}$.

corresponds to the bead size, good miscibility can be observed (cf. Figure 1.5B). With the decreased bond length $b_0 = 0.20\,\text{nm}$ the two-bead molecules tend to aggregate, due to the increased density of interactions (cf. Figure 1.5B).

Chapter 2

Quantifying membrane curvature sensing of peripheral proteins by simulated buckling and umbrella sampling

Kai Steffen Stroh, Herre Jelger Risselada

Reprinted with permission from
Journal of Chemical Theory and Computation,
2021, volume 17, issue 8, pp. 5276-5286.
DOI: 10.1021/acs.jctc.1c00021.

https://pubs.acs.org/articlesonrequest/AOR-MZ7XJYKWIQX2XTV5UKMF

Abstract

Membrane curvature plays an essential role in the organization and trafficking of membrane associated proteins. Comparison or prediction of the experimentally resolved protein concentrations adopted at different membrane curvatures requires direct quantification of the relative partitioning free energy. Here, we present a highly efficient and simple to implement free energy calculation method which is able to directly resolve the relative partitioning free energy of proteins as a direct function of membrane curvature, i.e. a curvature sensing profile, within (coarse-grained) molecular dynamics simulations. We demonstrate its utility by resolving these profiles for two known curvature sensing peptides, namely ALPS and α-synuclein, for a membrane curvature ranging from -1/6.5 to +1/6.5 nm^{-1}. We illustrate that the difference in relative partitioning (binding) free energy between these two extrema is only about 13 $k_{\mathrm{B}}T$ for both peptides, illustrating that the driving force of curvature sensing is subtle. Furthermore, we illustrate that ALPS and α-synuclein sense curvature via a contrasting mechanism, which is differentially affected by membrane composition. In addition, we demonstrate that the intrinsic spontaneous curvature of both of these peptides lies beyond the range of membrane curvature accessible in micropipette aspiration experiments, being about 1/7 nm $^{-1}$. Our approach offers an efficient and simple to implement *in silico* tool for exploring and screening the membrane curvature sensing mechanisms of proteins.

Contents

2.1 Introduction

Membrane curvature is an ubiquitous feature of cells. On one hand proteins can shape membranes. This curvature generation plays an important role in membrane fusion and fission. On the other hand membrane curvature sensing can control protein function by providing spatial information, thus providing a cue when and where biological processes are happening [1, 2]. Membrane curvature-dependent biomechanical coupling mechanisms may thus contribute to the sorting, trafficking and function of transmembrane proteins and membrane associated proteins[3–9]. Membrane binding proteins have been observed to sort toward membrane regions with a positive curvature [10–12], negative curvature [13], and even micrometer-scale membrane curvature (flat membranes) [14]. Furthermore, membrane proteins may even dynamically alter their curvature preference upon external stimuli. For example, the curvature mediated sorting of G protein-coupled receptors (GPCRs) is dynamically and differentially regulated in living cells via ligand-specific binding [6, 15]. Sensing of negative [13] or micrometer scale curvature [14] is facilitated by a fixed, effective overall protein shape which matches such a curvature, analogous to the sensing (and generation) of positive curvature by scaffold proteins such as banana-shaped bar domains [16]. In contrast, amphipathic protein motifs such as the ALPS motifs as well as the protein α-synuclein rather sense positive membrane curvature because of the concomitant increase in hydrophobic lipid packing defects [9].

The driving force of curvature-mediated protein sorting (the sorting force) is the minimization of the relative free energy of membrane adhesion. A quantitative relationship between membrane curvature and the concomitant relative free energy of membrane adhesion therefore yields valuable information regarding a protein's ability to sense membrane curvature. However, such a detailed quantification of sensing requires a large continuous range of different membrane curvatures. Conveniently, uni-dimensional lateral compression of a bilayer results in a stable buckled membrane whose curvature gradually ranges from highly negative to highly positive and whose shape can be analytically derived [17]. Previous computational approaches exploited a buckled membrane to test a protein's preference for membrane curvature [18–21]. However, unbiased MD simulations, being restricted to thermal fluctuations, often only sample the region close to the resolved free energy minimum. Although this suffices to quantify the preferred curvature of a protein (or lipid), if one is additionally interested in free energy differences and sorting forces over a wider range of curvatures, enhanced or biased sampling is essential. In particular, these insights would uniquely enable quantitative *in silico* screening of proteins and protein mutation on curvature sensing properties as well as furthering the rational design of proteins. In addition, quantifying differences in curvature dependent membrane adhesion free energy would enable direct prediction (and comparison) of the protein concentration differences measured in, for example, micropipette-aspiration experiments [22].

Umbrella sampling requires the selection of a suitable reaction coordinate and a bias potential that restrains the system to a desired state along the reaction coordinate. Because of the small magnitude of the sorting force, relatively long sampling times are needed within a large system (e.g., $20^3 = 8000\,\mathrm{nm}^3$ or more). Therefore, calculation of the bias potential must be computationally efficient, i.e. it should not hamper the optimization of simulation engines. Our general approach is to first calculate the relative free energy along the arc length s of the buckled membrane. Membrane undulations are suppressed to ensure a unique and well-defined mapping from s to local curvature, being a collective property of lipids. Then, a Jacobian transformation is performed to reconstruct the concomitant free energy as a function of local membrane curvature. This final step only relies on highly efficient and easily implemented pre- and post-processing steps.

Here, we will illustrate how a buckled membrane setup can be used in combination with umbrella sampling to efficiently obtain a direct and quantitative relationship between local membrane curvature and the protein's relative adhesion free energy – the curvature sensing profile of a protein. To this aim, we will resolve the curvature sensing profiles of two known curvature sensing peptides, namely ALPS and α-synuclein, within a curvature ranging from -1/6.5 to +1/6.5 nm^{-1}. We illustrate that the difference in relative adhesion free energy between these two extrema is only about $13\,k_\mathrm{B}T$ for both peptides, illustrating that curvature sensing is rather subtle. Importantly, both sensing profiles do not reveal the existence of a free energy minimum within the studied range. This suggests that the actual intrinsic spontaneous curvature of these peptides lies beyond the range of membrane curvature accessible in micropipette aspiration experiments, being about 1/7 nm^{-1}[23].

2.1.1 Analytical shape of buckled membranes

In the Helfrich framework the bending energy of a lipid bilayer can be expressed as a functional of its surface shape contour S [24]

$$F[S] = \int_S \mathrm{d}A \left(\frac{1}{2}\kappa (K(S) - K_0)^2 + \bar{\kappa} K_\mathrm{G}(S) \right) \tag{2.1}$$

with the mean and the Gaussian curvature modulus, κ and $\bar{\kappa}$, the total curvature K and the Gaussian curvature K_G. K_0 denotes the spontaneous bilayer curvature.

Following the work of Hu et al.[17] we describe the membrane shape by the angle $\psi(s)$ as a function of arc length parameter s (Fig. 2.1). Assuming the membrane lies in the xy-plane of a box with side lengths (L_x, L_y, L_z), and the buckle undulation follows the x-direction, the energy can be written as

$$F[\psi(s)] = \frac{L_y \kappa}{2L} \int_0^1 \mathrm{d}s (\psi'(s) - LK_0)^2 \tag{2.2}$$

where L is the membrane contour length along the buckle and $\psi'(s) = KL$. The compressive strain γ relates the contour or uncompressed length L to the projected length L_x

$$\gamma = 1 - L_x/L \tag{2.3}$$

Minimizing F while keeping the projected length $L_x = L \int_0^1 \mathrm{d}s \cos(\psi(s))$ fixed leads to

$$\frac{\partial^2 \psi(s)}{\partial s^2} = -\lambda_L \sin(\psi(s)) \tag{2.4}$$

with the Lagrange multiplier λ_L. The analytical solution of Eq. 2.4 leads to Jacobi elliptic functions. Integration of the cosine and sine of $\psi(s)$ yields the cartesian coordinates

$$X(s) = \int_0^s \mathrm{d}\sigma \cos(\psi(\sigma)), \tag{2.5a}$$

$$Z(s) = \int_0^s \mathrm{d}\sigma \sin(\psi(\sigma)) \tag{2.5b}$$

of the buckled shape. Starting from here Elías-Wolff et al. developed the following approach that we use in our work and summarize here for the sake of completeness [19]. To avoid the use of Jacobi elliptic functions, the shape equations (2.5) are numerically solved for $\psi(s)$ with the boundary conditions

$$\psi(0) = \psi(L) = 0, \; X(0) = Z(L = 0), \; X(L) = L_x. \tag{2.6}$$

The numerical solution is then approximated by a Fourier series:

$$X_M(s,\gamma) = L_x[s + \sum_{n=1}^{M} a_n^{(x)}(\gamma)\sin(4\pi n s)] \tag{2.7a}$$

$$\begin{aligned} Z_M(s,\gamma) = {} & L_x[a_0^{(z)}(\gamma) \\ & + \sum_{n=1}^{M} a_n^{(z)}(\gamma)\cos(2\pi(2n-1)s)] \end{aligned} \tag{2.7b}$$

where the coefficients $a^{(x)}(\gamma)$, $a_n^{(x)}(\gamma)$ are obtained by fitting to the numerical solution. M is the number of series elements used. Lookup-tables are constructed for range $0 \le \gamma \le 0.85$ and interpolated by splines. [19] For later use, we write the curve coordinates as

$$\boldsymbol{X} = (x_0 + X(s,\gamma), z_0 + Z(s,\gamma))^T \tag{2.8}$$

with the offsets x_0 and z_0.

The actual fit minimizes the distances of the lipids' last tail beads to the theoretical shape description:

$$\min_{x_0,z_0,\gamma,s_j} \chi^2 = \min_{x_0,z_0,\gamma,s_j} \frac{1}{2}\sum_{j=1}^{N}[(x_0 + X(s_j;\gamma) - x_j)^2 + (z_0 + Z(s_j;\gamma) - z_j)^2] \tag{2.9}$$

Where (x_j, z_j are the coordinates, and N is the number of beads included in the fitting procedure.

The position of a peptide or lipid along the arc length is given by the projection of its center of mass coordinate $(x_\mathrm{p}, z_\mathrm{p})$ onto the curve $\boldsymbol{X}$:

$$s_\mathrm{p} = \arg\min_s[(x_0 + X(s,\gamma) - x_\mathrm{p})^2 + (z_0 + Z(s,\gamma) - z_\mathrm{p})^2] \tag{2.10}$$

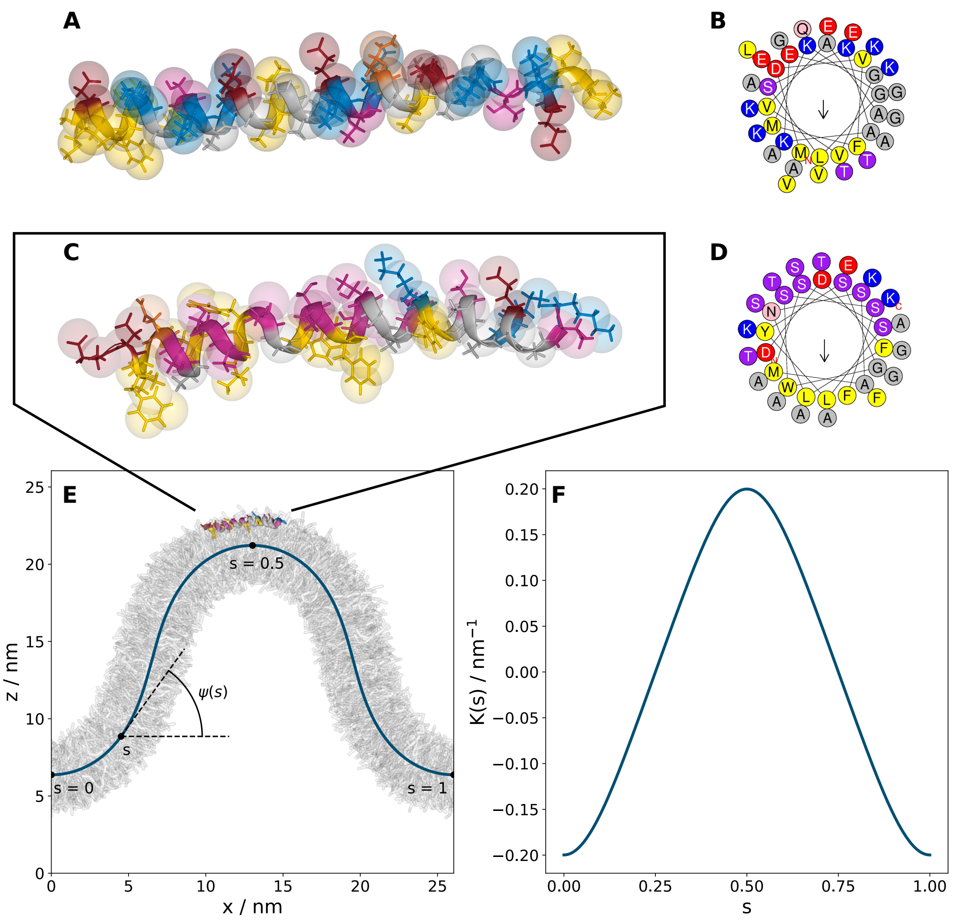


Figure 2.1: A: Superposition of the coarse-grained and backmapped [25] atomistic structure of aa 1-38 of α-synuclein, yellow: hydrophobic residues, violet: polar residues (visualized with nglview [26]) B: Wheel representation of aa 1-38 of α-synuclein (generated with HELIQUEST [27]) C: Superposition of the coarse-grained and backmapped atomistic structure of ALPS. D: Wheel representation of ALPS. E: Analytical membrane shape for a compression $\gamma = 0.39$, with lipids in grey. F: Midplane curvature as a function of arc length parameter s for the same compression.

With the sign convention that curvature is positive when the membrane bends away from the peptide, midplane curvature is given by:

$$K(s) = \frac{Z'(s)X''(s) - X'(s)Z''(s)}{(X'(s)^2 + Z'(s)^2)^{3/2}}$$

2.1.2 Umbrella sampling on a buckled membrane

Our goal is to calculate the free energy profile in the full curvature range of the buckled membrane. In principle, assuming ergodicity, the free energy or potential of mean force (PMF) can be extracted from MD simulations by monitoring the probability distribution $P(\xi)$ [28]:

$$F(\xi) = -k_{\mathrm{B}}T \ln\left[P(\xi)\right] + \text{const.} \tag{2.11}$$

As regions high in free energy are typically not well sampled in unbiased simulations we employ umbrella sampling [29], i.e., we split the reaction coordinate ξ into a number of windows and apply a biasing potential to restrain the system to a reference value of ξ. The typical choice is a harmonic potential

$$w_i(\xi) = \frac{k}{2}(\xi - \xi_i^{\mathrm{ref}})^2. \tag{2.12}$$

Where k is the force constant of the harmonic potential. A biased simulation yields the distribution $P_i^{\mathrm{b}}(\xi)$ and the unbiased free energy of each window can be calculated in the following way[29]:

$$F_i^{\mathrm{u}}(\xi) = -k_{\mathrm{B}}T \ln\left[P_i^{\mathrm{b}}(\xi)\right] - w_i(\xi) + F_i \tag{2.13}$$

The unknown constants F_i have to be calculated to combine the free energies of different windows. This is often done by using the weighted histogram analysis method (WHAM)[30]. The estimation of F_i can also be avoided, by calculating the unbiased mean force $\partial F_i^{\mathrm{u}}/\partial\xi$ instead of F_i^{u}. This approach is called umbrella integration (UI)[31].

An important factor in umbrella sampling is the selection of a suitable reaction coordinate. In our system, the obvious choice would be the arc length parameter s, since the shape description is based on that. This would mean evaluating the projection s_{p} to calculate the biasing potential

$$w_i(s_{\mathrm{p}}) = \frac{k_s}{2}(s_{\mathrm{p}} - s_i^{\mathrm{ref}})^2 \tag{2.14}$$

at every MD step. While this could be done by extending the simulation package of one's choice or by using additional software like PLUMED [32], the choice of s as the reaction coordinate is not trivially implemented and would introduce additional computational overhead, since the projection operation involves nonlinear optimization. We therefore decided to go a different route and use a local linear approximation instead. That is, for a given s_i^{ref} we calculate the cartesian reference coordinates $\boldsymbol{X}_i^{\mathrm{ref}}$ and the tangent $\hat{\boldsymbol{t}}_i$, cf. Fig. 2.2, and apply the umbrella potential along the tangent direction. Since we inhibit membrane fluctuations during umbrella sampling, the membrane shape properties remain constant. In particular, suppressing fluctuations ensures that the tangent directions are time independent. Otherwise, correctly applying the bias potential would necessitate a refit of the membrane shape at every MD step, rendering time integration prohibitively slow.

The following derivation is based on the work of Song and Zhu [33]. With the curve coordinates in the xz-plane described by

$$\boldsymbol{X} = (x_0 + X(s), z_0 + Z(s))^T \tag{2.15}$$

and

$$\boldsymbol{X}_i^{\text{ref}} \equiv \boldsymbol{X}(s_i^{\text{ref}}) \tag{2.16}$$

the linear expansion of the curve at a reference value s_i^{ref} has the form

$$\boldsymbol{X}(s_i^{\text{ref}} + \Delta s) \approx \boldsymbol{X}_i^{\text{ref}} + \Delta s \frac{\mathrm{d}\boldsymbol{X}_i^{\text{ref}}}{\mathrm{d}s}. \tag{2.17}$$

With the unit tangent vector

$$\hat{\boldsymbol{t}}_i \equiv \frac{\mathrm{d}\boldsymbol{X}_i^{\text{ref}}}{\mathrm{d}s} / L \tag{2.18}$$

The projection onto the linearized arc length parameter s can be written as

$$s_{\text{p}} = s_i^{\text{ref}} + (\boldsymbol{X}_{\text{p}} - \boldsymbol{X}_i^{\text{ref}}) \cdot \hat{\boldsymbol{t}}_i \tag{2.19}$$

This transforms the nonlinear optimization problem in Eq. (2.10) to a linear operation.

The bias potential now takes the form

$$w_i(\boldsymbol{X}_{\text{p}}) = \frac{k}{2} \left[(\boldsymbol{X}_{\text{p}} - \boldsymbol{X}_i^{\text{ref}}) \cdot \hat{\boldsymbol{t}}_i \right]^2. \tag{2.20}$$

with

$$k = k_s / L^2 \tag{2.21}$$

This way, we only rely on GROMACS' built in functionality of restraining the probe-molecule motion along a prescribed direction and our own pre- and post-processing. Thus, this approach is easily transferable to other simulation packages and has minimal additional computational cost.

An overview of a typical umbrella sampling setup with reference positions and corresponding tangents is given in Fig. 2.3.

After obtaining the free energy as a function of arc length parameter s one needs to apply the following variable transformation [34]:

$$F(K) = F(s) + k_{\mathrm{B}}T \ln \left[\left| \frac{\partial K}{\partial s} \right| \right], \tag{2.22}$$

that finally yields the binding free energy as a function of curvature. The full derivation of this change of variable of a PMF is given in the SI. This Jacobian correction accounts for the fact that in differently curved regions, traveling a certain distance along s corresponds to different amounts of curvature change. Obviously, the derivative $\partial K/\partial s$ becomes zero at the extrema of $K(s)$, that means as s approaches 0; 0.5; or 1, the logarithmic term in Eq. (2.22) tends to minus infinity and $F(K)$ cannot be calculated in this way at these points.

We note that for a probe whose center of mass is not close to the membrane midplane, e.g. a peripheral protein, the linear approximation to the midplane introduces a systematic error dependent on the sign and magnitude of the curvature, as the arc length travelled on the membrane surface is not the same as on the midplane. For negative curvature a certain distance along the surface corresponds to a larger distance in s. For positive curvature the opposite is true. This results in an under-/overestimation of the free energy in the respective regions.

To avoid this problem one can introduce a *parallel* or *offset curve* and apply a transformation similar to eq. (2.22).

The offset $\boldsymbol{X}_{\mathrm{o}}(s)$ is defined by

$$\boldsymbol{X}_{\mathrm{o}}(s) = \boldsymbol{X}(s) + d \cdot \hat{\boldsymbol{n}}(s) \tag{2.23}$$

with the signed distance d and the unit normal vector $\hat{\boldsymbol{n}}(s)$ [35].

The offset's derivative can be expressed in terms of properties of the generator curve $\boldsymbol{X}(s)$:

$$\boldsymbol{X}_{\mathrm{o}}' = (1 + K\,d)\boldsymbol{X}' \tag{2.24}$$

With this we can immediately write down the unit tangent and normal vectors to the offset curve

$$\hat{\boldsymbol{t}}_\mathrm{o} = \frac{1 + K\,d}{|1 + K\,d|}\hat{\boldsymbol{t}} \tag{2.25a}$$

$$\hat{\boldsymbol{n}}_\mathrm{o} = \frac{1 + K\,d}{|1 + K\,d|}\hat{\boldsymbol{n}} \tag{2.25b}$$

and we see that both curves are always parallel in membranes under consideration, as $1 + K\,d$ cannot become negative without the membrane touching itself.

The arc length between two points on the offset curve is:

$$\xi(s_2) - \xi(s_1) = \int_{s_1}^{s_2} |\boldsymbol{X}'_\mathrm{o}|\mathrm{d}s \tag{2.26a}$$

$$= \int_{s_1}^{s_2} |1 + K\,d|\,|\boldsymbol{X}'|\mathrm{d}s \tag{2.26b}$$

$$= \int_{s_1}^{s_2} |1 + K\,d| L\,\mathrm{d}s \tag{2.26c}$$

When the offset curve is the reaction coordinate a similar transformation to Eq. 2.22 has to be applied to obtain $F(s)$.

$$F(s) = F(\xi) + k_\mathrm{B}T \ln\left[\left|\frac{\partial s}{\partial \xi}\right|\right] \tag{2.27}$$

with

$$\left|\frac{\partial s}{\partial \xi}\right| = |1 + K\,d|^{-1}L^{-1} \tag{2.28}$$

For the midplane $\frac{\mathrm{d}s}{\mathrm{d}\xi} = L^{-1}$ is a constant and such a transformation would simply shift the whole free energy profile up or down, and since the free energy is only defined up to an undetermined constant, this can be omitted.

Finally, for going directly from ξ to midplane curvature, we have:

$$F(K) = F(\xi) + k_\mathrm{B}T \ln\left|\frac{\partial K}{\partial \xi}\right| \tag{2.29a}$$

$$= F(\xi) + k_\mathrm{B}T \ln\left|\frac{|1 + K\,d|^{-1}}{L}\frac{\partial K}{\partial s}\right| \tag{2.29b}$$

Throughout the paper, we exclusively use midplane curvature, if not stated otherwise. Alternatively, one could use the curvature of the membrane surface.

With the offset curvature expressed in terms of midplane curvature [35]

$$K_\text{o} = \frac{K}{|1 + Kd|} \tag{2.30}$$

we can immediately write down the free energy as a function offset curvature

$$F(K_\text{o}) = F(K) + k_\text{B}T \ln\left[\frac{1}{|1 + Kd|^2}\right]. \tag{2.31}$$

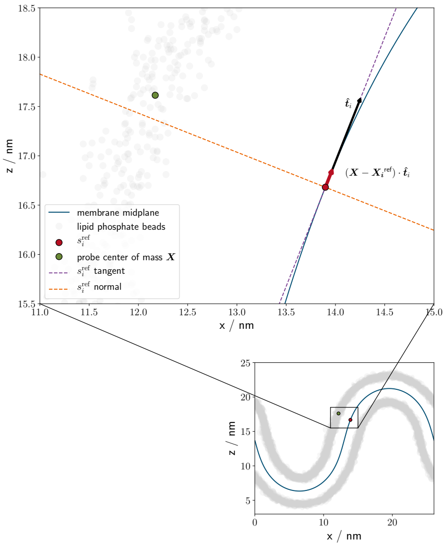


Figure 2.2: Local linear approximation of the reaction coordinate. The difference between probe and reference position is projected onto the tangent of the membrane midplane at s_i^ref. It is clearly visible that for small fluctuations around the reference position, the error due to the linearization is small. An increasing error in strongly curved regions could be further improved by adapting the force constant of the bias potential, i.e., a large k for strongly curved regions and smaller k in flatter regions.

2.1.2.1 Umbrella integration

To analyze umbrella sampling data, we use the aforementioned UI method [31]. UI is less reliant on window overlap than WHAM [28] and therefore allows more flexibility when trading off number of windows vs. error size [36].

In curvature sensing, we generally expect smooth and shallow free-energy profiles, without small barriers and UI is expected to perform optimally within such a

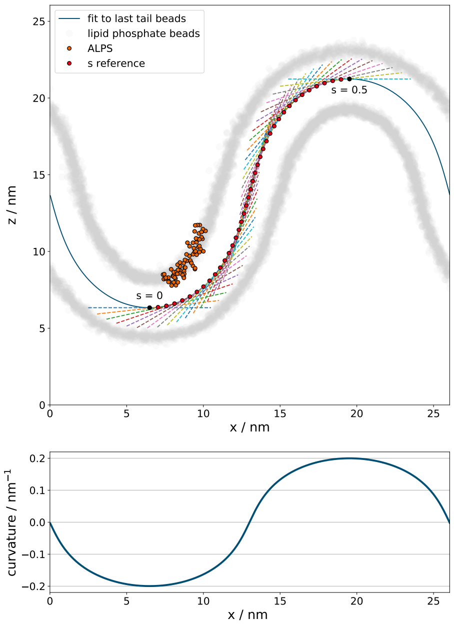


Figure 2.3: Umbrella sampling setup. The upper figure gives an overview of a typical setup. To generate the initial configurations the protein is pulled along the membrane surface. From this pulling trajectory frames are selected where the projection of the protein's COM onto the arc length is closest to the selected s-values. The umbrella potential acts along the tangent (dashed lines) at the selected arc length.
The lower figure shows the membrane midplane curvature as a function of the x-coordinate.

scenario. If the windows are chosen such that there is ample overlap, WHAM can be used just as well. In this case, comparisons of both methods have shown no significant difference.

The biased probability distribution P_i^{b} is approximated by a normal distribution

$$P_i^{\mathrm{b}}(\xi) = \frac{1}{\sqrt{2\pi}\sigma_i^{\mathrm{b}}} \exp\left[-\frac{1}{2}\left(\frac{\xi - \overline{\xi_i^{\mathrm{b}}}}{\sigma_i^{\mathrm{b}}}\right)^2\right], \tag{2.32}$$

which is well justified in the case of smooth free-energy profiles [28]. The mean $\overline{\xi_i^{\mathrm{b}}}$ and the standard deviation σ_i^{b} are extracted from simulations. The mean force of a window is

$$\frac{\partial F_i^{\mathrm{u}}}{\partial \xi} = k_{\mathrm{B}} T \frac{\xi - \overline{\xi_i^{\mathrm{b}}}}{(\sigma_i^{\mathrm{b}})^2} - k\left(\xi - \xi_i^{\mathrm{ref}}\right) \tag{2.33}$$

Combining the local mean forces yields the global mean force

$$\frac{\partial F}{\partial \xi} = \sum_i p_i(\xi) \frac{\partial F_i^{\mathrm{u}}}{\partial \xi} \tag{2.34}$$

with

$$p_i(\xi) = \frac{N_i P_i^{\mathrm{b}}}{\sum_j N_j P_j^{\mathrm{b}}}. \tag{2.35}$$

Where N_i is the number of timesteps in window i. The global potential of mean force $F(\xi)$ can be obtained by numerical integration of Eq. (2.34).

Statistical error: To ensure a well-equilibrated system and get confidence limits of the mean $\overline{\xi_i^{\mathrm{b}}}$ and the variance $(\sigma_i^{\mathrm{b}})^2$, we employ a procedure similar to ref. [37]. This procedure consists of dividing the trajectory ξ_i into nonoverlapping segments to decorrelate the data and a series of statistical tests: i) Mann-Kendall test for trend [38] in the mean of the segments, ii) Mann-Kendall test for trend in the variance of the segments, iii) D'Agostino's K^2 test for normality [39] of the mean of the segments, iv) von Neumann test for serial correlation [40]. After adjusting the starting timestep and the segment length such that all test are passed, the segmented data yields an estimate of the variance of $\overline{\xi_i^{\mathrm{b}}}$, $\mathrm{var}(\overline{\xi_i^{\mathrm{b}}})$, and the variance of $(\sigma_i^{\mathrm{b}})^2$, $\mathrm{var}((\sigma_i^{\mathrm{b}})^2)$, caused by sampling [36].

With $\mathrm{var}(\overline{\xi_i^{\mathrm{b}}})$ and $\mathrm{var}((\sigma_i^{\mathrm{b}})^2)$ error propagation of Eq. (2.33) leads to [36]:

$$\mathrm{var}\left(\tfrac{\partial F_i^{\mathrm{u}}}{\partial \xi}\right) = \tfrac{1}{\beta^2 (\sigma_i^{\mathrm{b}})^4} \left(\mathrm{var}(\overline{\xi_i^{\mathrm{b}}}) + \tfrac{(\xi - \overline{\xi_i^{\mathrm{b}}})^2}{(\sigma_i^{\mathrm{b}})^4} \mathrm{var}((\sigma_i^{\mathrm{b}})^2) \right) \tag{2.36}$$

For the variance of the combined mean force one obtains

$$\mathrm{var}\left(\tfrac{\partial F}{\partial \xi}\right) = \textstyle\sum_i p_i^2 \,\mathrm{var}\left(\tfrac{\partial F_i^{\mathrm{u}}}{\partial \xi}\right) \tag{2.37}$$

where variations of the weights are neglected [36].

Integrating over an interval $[\xi_a, \xi_b]$ and accounting for correlations between $\partial F / \partial \xi$ of different bins, finally yields [36]:

$$\mathrm{var}(\Delta F) \approx \overline{\mathrm{var}\left(\frac{\partial F}{\partial \xi}\right)} \left[(\xi_b - \xi_a) \sigma^{\mathrm{b}} \sqrt{2\pi} - 2(\sigma^{\mathrm{b}})^2 \right] \tag{2.38}$$

Where $\overline{\mathrm{var}(\partial F / \partial \xi)}$ is averaged over the interval $[\xi_a, \xi_b]$. Likewise σ^{b} is the average of σ_i^{b} over all windows in that interval.

2.1.3 Simulation details

Initial configurations were generated with the python script *insane.py* [41]. Approximately 675 lipids per leaflet (for details see Tab. 2.1) were put in the x-y plane of a 40 nm · 10 nm · 20 nm simulation box. The ALPS peptide was placed close ($\Delta z \approx 1$nm) to the upper leaflet. Subsequently the system was solvated with standard Martini water and a 0.15 M NaCl concentration.

The initial x-dimension can be calculated for a given compression γ and minimal curvature radius[1] by [17]

$$L = 4\pi R_{\min}\sqrt{\gamma}\left[1 + \frac{3}{16}\gamma + \frac{39}{512}\gamma^2 + \frac{303}{8192}\gamma^3 ...\right] \tag{2.39}$$

After steepest-descent energy minimization and initial equilibration (10 ns NVT and 50 ns NPT) the system was compressed in x-direction by applying a pressure of 3 bar in x-direction and allowing the system to expand in z-direction only (the y-dimension was kept fixed) using the Berendsen barostat ($\tau_p = 12.0$ ps, compressibility of $3 \cdot 10^{-4}$ bar^{-1}). From this compression-trajectory, the frame closest to the chosen compression was selected. This chosen frame, then served as starting configuration for another equilibration run using the Parrinello-Rahman barostat ($\tau_p = 12.0$ ps, compressibility of $3 \cdot 10^{-4}$ bar^{-1}). This equilibration phase is up to 5µs long. This additional equilibration serves several purposes: i) In systems with inhomogeneous membrane composition, a curvature-induced partitioning of lipids is observed, governed by the lipids' intrinsic molecular shape [20, 42]. With a comparatively long simulation time, we ensure that lipid resorting is finished. ii) A buckled membrane fluctuates around its analytical shape. For umbrella sampling the membrane will be restrained to that analytical shape. To minimize disturbance of the system we select the frame with the smallest deviation from the optimal shape and then restrain large scale movements of the lipids in the lower leaflet, such that the analytical shape is exactly maintained. For this purpose, position restraints with a small force constant (10 kJ $\cdot$ mol^{-1} $\cdot$ nm^{-2}) are applied to the headgroup (PO4 beads) of lower leaflet lipids. This way, influence on upper leaflet dynamics is minimized.

In the case of α-synuclein, an equilibrated configuration of the ALPS/membrane-system was taken, the ALPS peptide was deleted and α-synuclein inserted into the free volume. This was followed by energy minimization and NVT equilibration.

Initial configurations for the umbrella sampling were generated by pulling the protein along the membrane. All umbrella sampling runs are 1.05 µs long, with

[1] corresponding to the maximal curvature

the first .05 µs for equilibration. For ASYN PC55PE30PA15 sampling was subsequentially extended by 1 µs. We used harmonic potentials with a force constant of $k = 100\,\mathrm{kJ\,mol^{-1}\,nm^{-2}}$ to restrain the peptide to a defined point along the reaction coordinate.

All simulations were coupled to a constant heat-bath using the V-rescale algorithm [43]. The Umbrella sampling simulations were performed in the NVT ensemble in order to fully conserve the derived analytical shape of the buckle. It should be noted, however, that these umbrella simulations were pre-equilibrated within the NPT ensemble at a hydrostatic pressure of 1 bar (coupling to the z-direction).

Simulations were performed with GROMACS 2019.3 [44, 45] and GROMACS 2020.3 [46] using the Martini 2.2 force field [47–49]. Data analysis was performed with an in-house Python module, based on MDAnalysis [50, 51]. The shape fitting procedure is based on mdxrfile [52].

2.1.3.1 Protein models

The here-used model of ALPS is also used in ref. [53] and is based on previous atomistic simulations [54]. Based on the peptide's hydrogen bonding pattern, a helical secondary structure was assigned to all residues according to the definition used by DSSP [55, 56]. The backbone angle parameters of the central region of the peptide (GWSSFTTG) were relaxed to MARTINI's default values for loop regions ($k = 20\,\mathrm{kJ\,rad^{-2}\,mol^{-1}}$, angle = 96°) to allow for some flexibility, which is believed to play a role in defect sensing [57].

The model of α-synuclein only included the N-terminal region (residue 1-38) of α-synuclein, i.e. its curvature sensing motif, conform with earlier work of Pranke et al [9]. We will refer to this section of α-synuclein as ASYN throughout the paper. Its secondary structure was based on the crystal structure of α-synuclein, as obtained by Rao et al [58]. MARTINI parameters were assigned based on the DSSP definition of secondary structure.

2.1.3.2 Lipid composition

The choice of lipids used in our study is motivated by the main phospholipids found within highly curved mitochondrial membranes [59]. We vary the lipid composition to model the following scenarios: (i) A pure POPC membrane, i.e. as simple as possible, (ii) A negatively charged POPC/POPS membrane with uniform charge spatial distribution along the arc-length of the buckled membrane conform with the experiment work of Pranke. et al [9], and (iii) POPC/POPE/POPA negatively charged with a non-uniform spatial charge distribution, i.e., POPA preferentially sorts to regions with a negative membrane curvature. A detailed overview of the here-studied membrane setups is illustrated in table. 2.1

	# lipids per leaflet							umbrella sampling	
system	POPC	POPE	POPA	POPS	# water	# NA	# CL	# win.	time / µs
ALPS PC100	676	-	-	-	51507	567	567	60	1.0
ALPS PC55PE30PA15	371	202	101	-	51696	670	468	56	1.0
ALPS PC40PS60	270	-	-	405	51515	970	162	60	1.0
ASYN PC100	676	-	-	-	51507	565	567	60	1.0
ASYN PC55PE30PA15	371	202	101	-	51696	668	468	60	2.0
ASYN PC40PS60	270	-	-	405	51515	972	162	60	1.0

Table 2.1: System details. Membranes are labeled by the percentage of their lipid constituents. Upper and lower leaflets have equal lipid compositions. Number of windows used in umbrella sampling and sampling time per window is also given.

2.2 Results

Contrasting curvature sensing mechanisms of ALPS and α-synuclein can be distinguished.

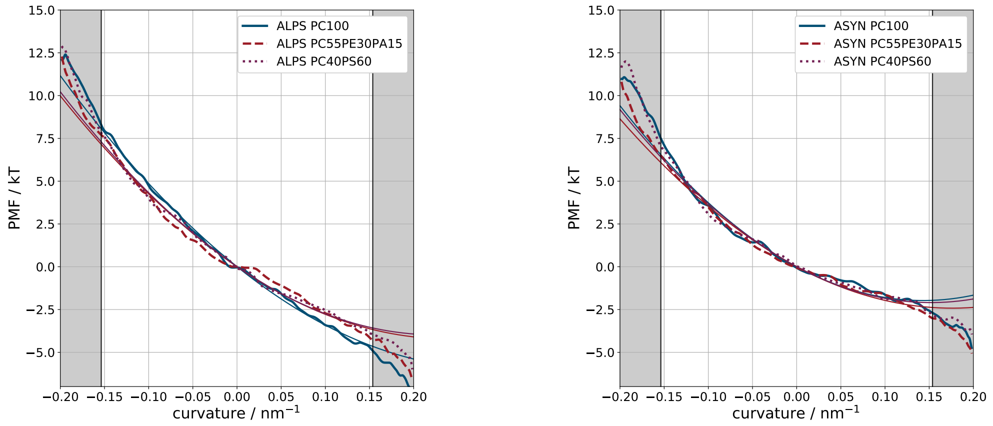
Figure 2.4 illustrates the binding free energy of ALPS and ASYN as a function of membrane curvature for different membrane compositions. For both peptides, the free energy difference between the two extrema in analyzed membrane curvature, i.e. -1/6.5 to +1/6.5 nm^{-1}, is in the range of 13 $k_\mathrm{B}T$. Such a moderate free energy scale illustrates that curvature sensing is in practice mediated by a subtle driving force. The magnitude of the thermodynamic sorting force acting on the peptide depends on the spatial gradient of membrane curvature. In our setup, which features a strong spatial gradient, the typical average sorting force acting on a peptide within the simulation is about a few pN.

However, ALPS and ASYN have contrasting chemistries and curvature sensing is believed to be mediated by distinct sensing mechanisms [9]. ALPS is electrically neutral and features a hydrophobic interface consisting of bulky hydrophobic residues, in contrast ASYN features a positive net charge with a "poorly developed" hydrophobic interface. Indeed, our simulation protocol is able to discern these mechanistic differences. In case of an electrically neutral membrane, the slope of the obtained binding free energy profile – sensing profile – is shallower for ASYN than ALPS within the positively curved region. Positive membrane curvature is accompanied by an increase in surface hydrophobicity due to the increased formation of lipid packing defects (cf. Fig. S7). As a consequence, owing to the poor hydrophobic face of ASYN, the concomitant sorting force is reduced with respect to ALPS. In contrast, introduction of negatively charged POPA lipids in the membrane affects the sensing of ASYN, due to its positive net charge, whereas the sensing behavior of ALPS is not affected. These negatively charged POPA lipids mainly accumulated in the negatively curved region (cf. Fig. S3), thereby alleviating the unfavorable hydrophobic interactions in that region. In contrast, such a curvature dependent accumulation of charged lipids is not observed for equivalent negatively charged POPS lipids. Consequently, the presence of PS lipids does not differentially affect the sensing behavior of ALPS and ASYN, despite increasing the absolute binding free energy of ASYN with respect to ALPS. The sorting behavior of charged proteins seems thus only modulated by charged lipid species with a preferential membrane curvature.

The spontaneous curvature of ALPS and α-synuclein lies below the experimentally accessible curvature range.

Interestingly, the absence of a free energy minimum in Fig. 2.4 indicates that there in fact exists no optimal preferential curvature for the peptide within our buckled membrane system. The peptide is attracted to the region of maximal positive membrane curvature only because of a spatially dependent free energy minimum. An even stronger positive curvature would in fact still be more favorable. This can be understood from the expected increase in surface hydrophobicity associated with a stronger positive membrane curvature. The curvature range in our membrane protocol, from -1/6.5 upto 1/6.5 nm^{-1}, well captures the curvature of the membrane tubes formed by α-synuclein in experiments, being 1/5 to 1/20 nm^{-1} [60]. In the limit of high peptide concentration, the relationship between the radius of a spontaneously formed membrane tube, R_s, and the peptide's spontaneous curvature, $\bar{K}_\mathrm{p}$, is given by, $R_s = (1 + \frac{\kappa}{\bar{\kappa}})\bar{K}_p^{-1}$, where κ is the bending modulus of the protein free membrane and $\bar{\kappa}$ an elastic constant describing the effective curvature coupling between peptide and membrane [23, 61]. Since $\bar{\kappa}$ has a positive value, $\bar{K}_p^{-1}$ is smaller than R_s. For BAR-domain protein families the inverse spontaneous curvature $\bar{K}_p^{-1}$ has been reported to be in the range of 5 nm or less [62–64], which is somewhat puzzling given the 20 nm-sized dimension of these protein complexes. Visual inspection (extrapolation) of the sensing profiles of ALPS and α-synuclein suggests similar small values for $\bar{K}_p^{-1}$, i.e. values that are similar or less than the hydrophobic thickness of the membrane itself (about 4 nm).

The membrane binding free energy of peptides is assumed to quadratically scale with curvature, $F = \int_A \phi \frac{\bar{\kappa}}{2}(K - \bar{K}_\mathrm{p})^2$, with ϕ being the peptide concentration [23]. Since $\phi \rightarrow 1/A$ in our example, the ϕ term effectively drops out when integrating over the membrane area A. Therefore, the peptide's spontaneous curvature $\bar{K}_\mathrm{p}$ and the coupling parameter $\bar{\kappa}$ can be extracted from the data. Our results (cf. Tab. 2.2) suggest a $\bar{K}_\mathrm{p}^{-1}$ of roughly 4 nm for ALPS. For ASYN on the two membranes with little or no gradient in lipid composition (PC40PS60 and PC100, respectively)



(a) Relative binding free-energy of ALPS as a function of curvature.

(b) Binding free-energy of first helix of α-synuclein as a function of curvature.

Figure 2.4: Relative binding free-energy as a function of curvature. The grey areas indicate curvature values that are present in the system but are not included in the final analysis. Anything below a radius of 6.5 nm is discarded. Close to the extrema several problems accumulate. PMF $\propto K^2$ is not valid any more. The linearization error increases and the $\ln|\frac{\partial K}{\partial s}|$ correction to the free energy breaks down.

fitting yields $\bar{K}_\mathrm{p}^{-1} \approx 7\,\mathrm{nm}$. In the case of the PC55PE30PA15 membrane, which has a notable curvature-dependence in lipid composition, the apparent $\bar{K}_\mathrm{p}^{-1}$ is reduced to 5.7 nm. Furthermore, the value of $\bar{\kappa}$ is estimated to be in the range of $200\,k_\mathrm{B}T$ (cf. Tab. 2.2), which is tenfold larger than the bending modulus of the membrane κ, about $20\,k_\mathrm{B}T$ [65]. This agrees with the notion that $\bar{\kappa} > \kappa$, which is essential for a membrane curvature inducing peptide [23].

These here-reported values indicate that $\bar{K}_p^{-1}$ lies below an experimentally accessible length scale. This implies that stronger membrane curvatures are always preferable in practice despite the existence of a finite preferential curvature. In addition, these extreme curvatures may simultaneously challenge the accuracy and validity of its underlying second-order continuum elastic approximation [65]. Finally, it should be noted that membrane tubulation can be alternatively understood within the concept of a protein induced spontaneous tension and a concomitant spontaneous membrane curvature, thereby excluding the need for an explicit description of $\bar{K}_\mathrm{p}^{-1}$ and $\bar{\kappa}$ [66, 67].

system	ΔF / $k_\mathrm{B}T$	$\bar{\kappa}$ / $k_\mathrm{B}T$	$\bar{K}_\mathrm{p}$ / nm^{-1}	$\bar{K}_\mathrm{p}^{-1}$ / nm
ALPS PC100	13.2 ± 1.4	143.9	0.29	3.5
ALPS PC55PE30PA15	12.0 ± 1.4	146.9	0.24	4.2
ALPS PC40PS60	11.4 ± 1.4	157.0	0.23	4.4
ASYN PC100	10.2 ± 1.4	193.5	0.14	7.0
ASYN PC55PE30PA15	9.5 ± 1.0	156.3	0.18	5.7
ASYN PC40PS60	9.9 ± 1.4	182.2	0.15	6.6

Table 2.2: Results from fitting the sensing profiles to the free energy $F \rightarrow \frac{\bar{\kappa}}{2}(K - \bar{K}_\mathrm{p})^2$. Uncertainties in free energy differences are calculated from the statistical error (cf. Eq. 2.38) and are given as the 95% percent confidence interval: $1.96\sqrt{\mathrm{var}(\Delta F)}$.

2.3 Discussion

Recent works have illustrated both the utility and versatility of buckled membranes to study the curvature preference of different membrane constituents [18–21, 42]. Here, we have illustrated how a buckled membrane can be combined with umbrella sampling techniques to calculate the relative binding free energy as a function of membrane curvature for curvature sensing peptides. This method is able to discern the distinct sensing mechanisms of ALPS and ASYN and enable *in silico* extraction of elastic properties such as the spontaneous curvature of the peptide and the elastic coupling modulus. The Jacobian transformation, required to reconstruct the relative binding free energy as a function of membrane curvature rather than the spatial coordinate s, becomes ambiguous near the curvature minima of the buckle. This challenges accurate quantitative extraction of binding free energies in unbiased simulations. In contrast, our biased simulation method circumvents this problem by sampling otherwise inaccessible regions where the Jacobian transformation can be accurately applied. In addition, this approach provides the obvious advantage that a wide continuous range of different curvatures can be studied within the same computationally efficient setup, in contrast to approaches that utilize membrane vesicles of a different size [13, 68].

Our method is sufficiently accurate to capture the subtle nature of curvature sensing. Error estimations suggest that the obtained free energy differences are within an error of $1.4\,k_\mathrm{B}T$. To reconstruct the potential of mean force, we used the aforementioned umbrella integration (UI) method [31]. UI is less reliant on window overlap than weighted histogram methods [28] and therefore allows us to reduce the number of umbrella windows and thus the number of performed simulations, albeit at the cost of potentially larger errors [36]. However, in curvature sensing, we generally expect smooth, monotonous and shallow free-energy profiles, without small barriers and UI is expected to perform optimally within such a scenario.

Peptides can display a curvature dependent preferential rotation angle (see Fig. S5). Rotation of the peptide is particularly restricted within regions of strong negative curvature. In this region the rotation angle is restricted to an approximately 40 rather than 360° range. In contrast, regions with a strong positive curvature enable full rotation of the peptide, but display a weak preferential orientation as evident from a non-uniform angle distribution. However, this rotational entropy contribution can be considered as an intrinsic property of membrane curvature and is therefore included in our definition of relative binding free energy. Hence, a similar cost in rotational entropy is expected when a peptide partitions on a membrane tube or tether in experiments, i.e. a cylindrical membrane surface that is subject to two different principal curvatures [23]. The associated free energy contribution due to a difference in rotational entropy is small in our example: Restriction of peptide rotation to a range of 40° results in a free energy cost of $\Delta F = -k_\mathrm{B}T \ln(40/360) = -2.2\,k_\mathrm{B}T$. Thus, the rotational free energy cost associated with partitioning in a region with a strong negative curvature is only about $2\,k_\mathrm{B}T$. For regions of positive membrane curvature – where only a weak preferred directionality exists – such a contribution is expected to become even smaller ($\leq k_\mathrm{B}T$A more detailed analysis of the relationships between peptide orientation angle and curvature sensing can be found in the *SI*.

Since all degrees of freedom but ξ are integrated out in the derivation of $F(\xi)$, sufficient sampling of the configuration space perpendicular to ξ has to be ensured, for the estimate of $F(\xi)$ to be accurate [28]. As we have discussed in the SI as an example, extending our umbrella sampling protocol with enhanced sampling methods such as Hamiltonian replica exchange [69, 70] is straightforward.

Incorporating enhanced sampling methods might become particularly useful when considering larger conformational rearrangements of proteins. E.g., for ALPS it is assumed that its curvature sensing ability involves changes in secondary structure[54]. As the Martini model cannot capture such changes, this aspect is not included in our present study. However, our protocol is independent of this force field choice.

In principle, such free energy estimations are feasible employing an all-atom (AA) force field, albeit at significantly higher computational effort. The rather weak sorting force $\partial F/\partial K$ entails relatively long sampling times. Compressing the membrane, initial equilibration, lipid resorting, and conformation generation or the umbrella sampling could still be done in a coarse-grained representation. The starting conformation of each window would be back-mapped to full atomistic detail. Optimizing the window spacing and the force constant of the umbrella potential, would be particular important for AA simulations.

Our buckling approach seems particularly suited for studying curvature sensing of peptides and smaller proteins or protein complexes of up to a size of 5 nm. Since the approach critically relies on a spatial gradient in membrane curvature, local membrane curvature is less accurately defined for wider protein complexes such as scaffold proteins. Moreover, the soluble parts of large protein complexes may actively interact with the membrane and thereby compromise accuracy, especially within the negative curvature region. This problem can be somewhat circumvented by increasing the length-scale of the buckled membrane. However, this would simultaneously result in a smaller sorting force therefore requiring additional sampling time.

Our current study focused on studying membrane binding proteins and peptides rather than transmembrane proteins such as GPCRs [6, 15]. In this work, the distal monolayer was restrained to conserve the shape of the membrane during the umbrella simulations and more accurately reconstruct local membrane curvature. To avoid artifacts caused by interactions of restrained lipids with transmembrane domains, shape conservation has to be achieved while enabling free diffusion in the whole bilayer, e.g., by employing a custom potential similar to a flat bottom potential or dummy particles that are placed on the membrane surface and interact with lipids only [71].

Finally, the here-presented method offers a unique possibility to directly compare curvature dependent partitioning free energies from simulations with experiments. In particular, the ability to screen proteins *in silico* for their potential curvature sensing properties while simultaneously quantifying enables a high throughput in comparison to present experimental approaches, which rely on labor intensive fluorescent labeling of proteins and/or careful tweaking of the setup with optical tweezers [23]. This time efficiency of simulations could be exploited to pin-point relevant regions in either chemical or protein space which are subsequently studied by experiments. GPCRs have recently been shown able to dynamically alter their curvature sensing properties upon ligand binding [6, 15]. Our approach may offer an efficient tool for directly exploring the curvature switching mechanisms of GPCRs and other proteins by molecular dynamics simulations.

Appendix 2.A Supporting Information

2.A.1 Relative binding free energy

Free energy profiles are shown in Fig. 2.5 as a function of reaction coordinate ξ and arc length parameter s. Since ξ represents an offset curve (cf. Fig. 2.6), a correction term $k_\mathrm{B}T \ln |\frac{\partial s}{\partial \xi}|$ is applied to transform $F(\xi)$ to $F(s)$. The offset distance d, that is used in the construction of the offset curve $\boldsymbol{X}_\mathrm{o}(s) = \boldsymbol{X}(s) + d \cdot \hat{\boldsymbol{n}}(s)$, is extracted from the umbrella sampling trajectories by

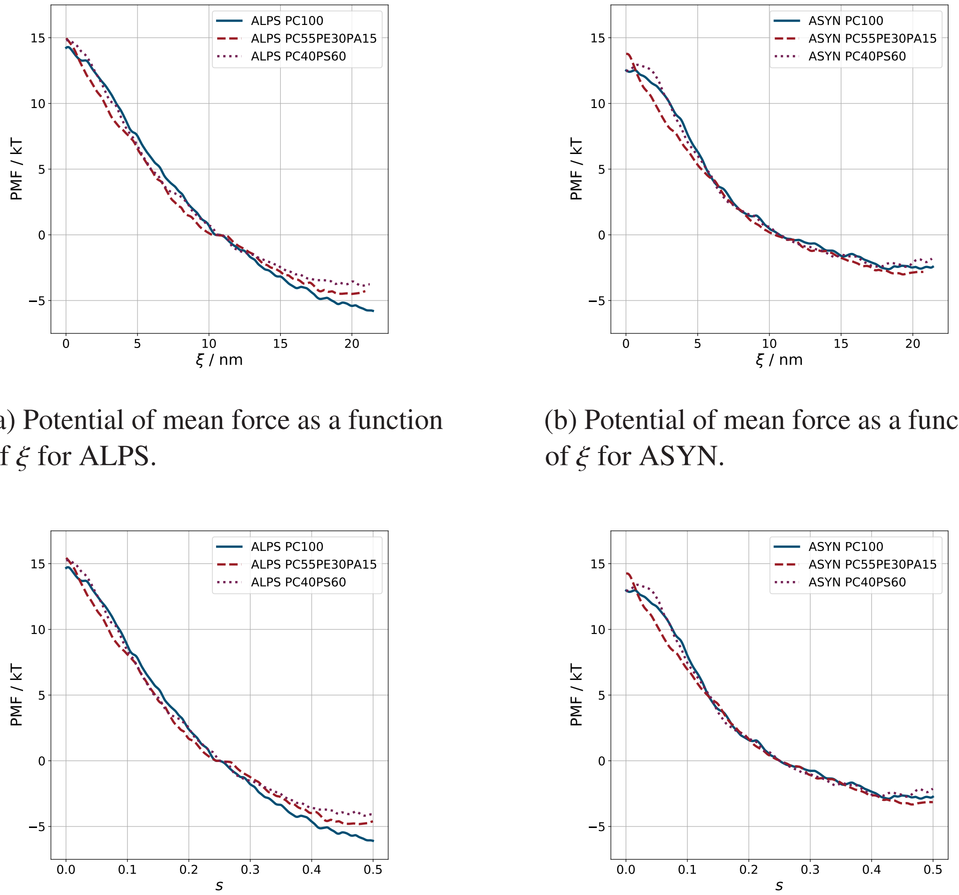
$$d = \frac{1}{N_\mathrm{win}} \sum_{i}^{N_\mathrm{win}} \left\langle (\boldsymbol{X}_\mathrm{p} - \boldsymbol{X}_i^\mathrm{ref}) \cdot \hat{\boldsymbol{n}}_i \right\rangle \tag{2.40}$$

with the peptide's center of mass $\boldsymbol{X}_\mathrm{p}$, the reference position on the midplane $\boldsymbol{X}_i^\mathrm{ref}$ of the i-th window, and the unit normal vector $\hat{\boldsymbol{n}}_i$. Table 2.3 lists the values of d for every system used.

system	d / nm
ALPS PC100	1.78 ± 0.12
ALPS PC55PE30PA15	1.91 ± 0.22
ALPS PC40PS60	1.80 ± 0.14
ASYN PC100	1.81 ± 0.17
ASYN PC55PE30PA15	1.91 ± 0.22
ASYN PC40PS60	1.84 ± 0.19

Table 2.3: Mean offset d in all simulated membrane systems.

A few general remarks about the representation of a potential of mean force (PMF) as a function of ξ or s: i) Due to the membrane's symmetry, free energy profiles are expected to be symmetric around $s = 0.5$. ii) For a peptide that has a preferred curvature K_p equal or larger than the maximum curvature of a given membrane, there will be a free energy minimum at $s = 0.5$. If a peptide's K_p is beyond what is available on the membrane, the sorting force will drive the peptide to the top of the buckle, where the mean force will be zero, i.e., $\frac{\partial F(s)}{\partial s}|_{s=0.5} = 0$. This does not automatically mean, that the relative binding free energy with respect to curvature has a minimum at this point. iii) In this representation, comparability of a peptide's curvature sensing ability on different membranes is not given in general. Membrane shape obviously influences the free energy profile. In our case membranes were constructed to have similar shape, but minor differences still exist. Membrane length differs slightly, as one can see in the plots of $F(\xi)$.



(a) Potential of mean force as a function of ξ for ALPS.

(b) Potential of mean force as a function of ξ for ASYN.

(c) Potential of mean force as a function of s for ALPS.

(d) Potential of mean force as a function of s for ASYN.

Figure 2.5: Potential of mean force. With $F(s) = F(\xi) + k_{\mathrm{B}}T \ln|\frac{\partial s}{\partial \xi}|$. The curves are shifted such that the free energy is 0 in the flat region of the membrane, i.e., $F(s = 0.25) = 0\,k_{\mathrm{B}}T$.

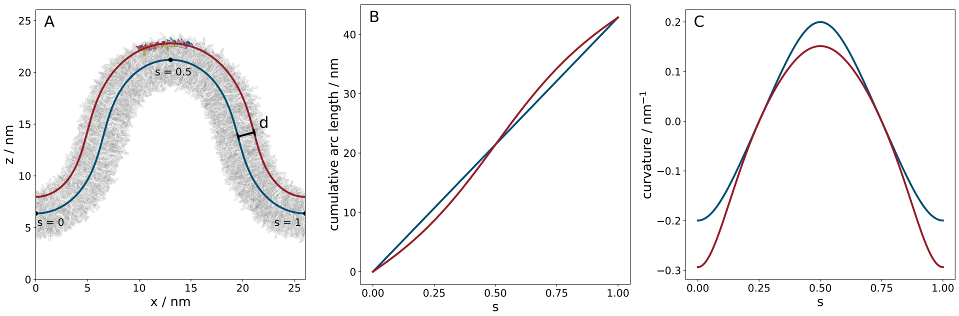


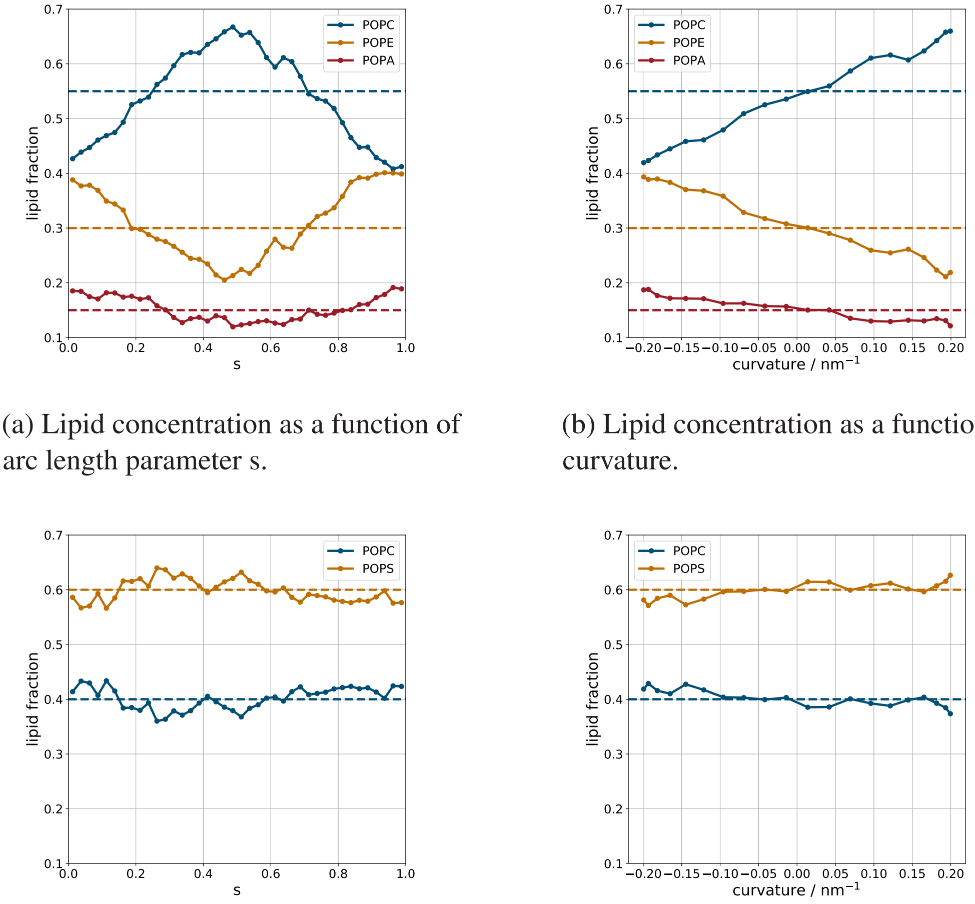
Figure 2.6: A: Midplane (blue) and offset curve (red). B: Cumulative arc length. C: Curvature.

2.A.2 Lipid partitioning

Figure 2.7 shows lipid partitioning of the upper layer. A lipids center of mass is projected onto the midplane. The obtained histograms are normalized by the total number of lipids in a specific bin. A pronounced curvature dependent sorting is observed in the PC55PE30PA15 membrane. Lipids with a small headgroup-to-tail volume ratio are increasingly found in the negatively curved region. This is in agreement with previous work [20, 42]. The accumulation of POPA to the negative region leads to an curvature-dependent surface charge of the membrane. The PC40PS60 membrane has a nearly constant lipid composition over the whole curvature range.

2.A.3 Peptide orientation

Peptide orientation is measured as the angle θ between the vector $\boldsymbol{p}$ from N- to C-terminus and the tangent $\boldsymbol{t}$ in the tangent plane (cf. Fig. 2.8) defined at the peptide's center of mass. In moderately curved regions, i.e., $s \in [0.2, 0.3]$, both peptides can freely rotate around the membrane normal (see Fig. 2.9). In strongly curved regions both peptides have distinct preferred orientations. As ASYN is one stiff helix, it is no surprise that it favors an orientation perpendicular to the buckling direction. ALPS, on the other hand, has a flexible middle section and therefore can adapt to the curved membrane. Our data suggests that at the very top of the buckle ($s = 0.5$) ALPS has two preferred orientations, namely $\theta \approx -45°$ and $\theta \approx +135°$. This kind of symmetry is expected, as the membrane is symmetric around $s = 0.5$, rotating the peptide by 180° is identical in terms of free energy. The fact that $\theta \approx +45°$ and $\theta \approx -135°$ are clearly less favorable, indicates that ALPS has some kind of asymmetry that favors one orientation relative to the top of the buckle. Therefore, when moving away from the top, only one of the states $\theta \approx -45°$ or $\theta \approx +135°$ will represent a minimum in the



(a) Lipid concentration as a function of arc length parameter s.

(b) Lipid concentration as a function of curvature.

(c) Lipid concentration as a function of arc length parameter s.

(d) Lipid concentration as a function of curvature.

Figure 2.7: Overall composition as indicated by the dashed vertical lines.

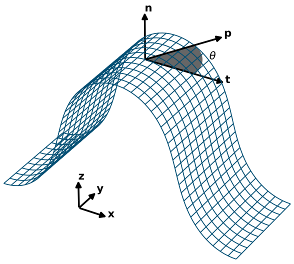
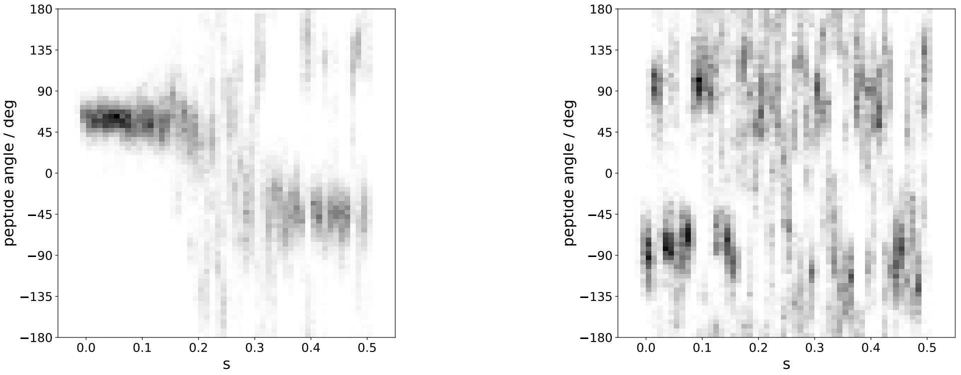


Figure 2.8: Cartoon illustrating the orientation of the peptide on the tangent plane.

rotational free energy landscape, which is what is observed.

2.A.4 Umbrella sampling with restrained rotation

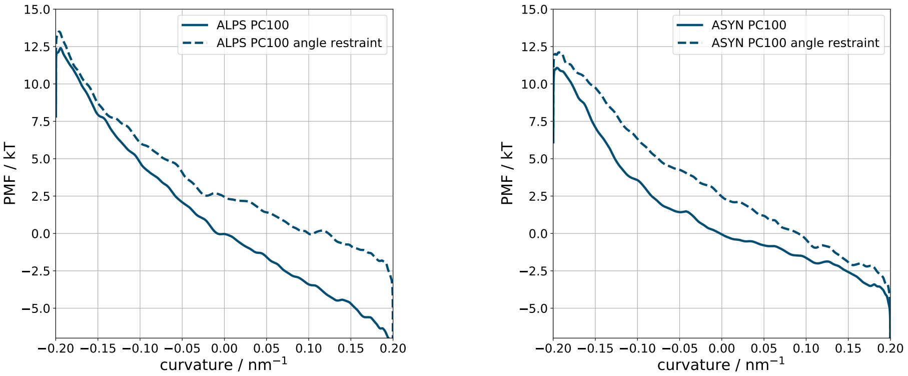
Due to the anisotropic nature of the buckled membrane, a peptide that is not perpendicular to the buckle direction samples a range of different curvature values at once. This can be avoided by forcing the peptide into a perpendicular orientation. For this purpose the peptides were restrained to a conformation parallel to the y-axis with GROMACS' pull code, using a force constant of $100\,\mathrm{kJ}\cdot\mathrm{mol}^{-1}\cdot\mathrm{rad}^{-2}$. Figure 2.10 shows a comparison of free energy profiles with ($F_{\mathrm{restr}}(K)$) and without ($F_{\mathrm{free}}(K)$) an angle restraint. Following a semi-quantitative line of reasoning, one can see that for ASYN (Fig. 2.10b) the differences of the free energy profiles are mostly due to changes in rotational entropy. As before, we define $F_{\mathrm{free}}(K = 0) = 0$ for the case of free rotation. As can be seen from Fig. 2.9, the peptide samples all possible angles in the flat region of the membrane. With the angle restraint in place, rotation is restricted to a range of roughly 30°, resulting in a free energy difference $\Delta F \approx -k_{\mathrm{B}}T\ln(30°/360°) = 2.5\,k_{\mathrm{B}}T$. Assuming the system is quasi-isotropic in the flat region, we expect no other contributions to the free energy difference, and set $F_{\mathrm{restr}}(K = 0) = 2.5\,k_{\mathrm{B}}T$. Around the curvature extrema, ASYN is oriented perpendicular to the buckling direction even without the angle restraint. Since, the number of available conformations is reduced by 1/2 by the rotational restraint (the peptide is parallel or antiparallel to the y-axis without the angle restraint and parallel with the angle restraint), the difference in rotational entropy results in a free energy difference of $-k_{\mathrm{B}}T\ln(1/2) = 0.7\,k_{\mathrm{B}}T$, which is consistent with the data shown in Figure 2.10b.



(a) ALPS orientation on the tangent plane as a function of arc length parameter s. The reported angle corresponds to the illustration in Fig. 2.8. For negative and positive curvature values (around $s = 0.0$ and $s = 0.5$, respectively) peptide rotation is diminished. In the flat region all orientations are sampled.

(b) ASYN orientation on the tangent plane as a function of arc length parameter s. For negative and positive curvature values (around $s = 0.0$ and $s = 0.5$, respectively) peptide rotation is diminished. In the flat region all orientations are sampled.

Figure 2.9



(a) ALPS with and without restrained rotation.

(b) ASYN with and without restrained rotation.

Figure 2.10

For ALPS the free energy difference (cf. Fig 2.10a) cannot be explained by the differences in rotational entropy. As discussed above and in section 2.A.5, on top of the buckle ALPS prefers an orientation of $\theta \approx -45°$ and $\theta \approx +135°$. From the probability distribution $P(\theta)$ of the replica exchange simulations (cf. Fig. 2.11b) it is clear that $\theta = \pm 90°$ is not a minimum of the free energy in rotation space $F(s = 0.5, \theta)$. The difference in frequency that these states are observed (cf. Fig. 2.11b), rather suggests a free energy difference of a few $k_{\mathrm{B}}T$ between $\theta = \pm 90°$ and $\theta \in \{-45°, 135°\}$. A detailed investigation of the underlying physical processes is beyond the scope of this paper. However, the results are in line with what is observed for LL-37 on a buckled membrane [18], an antimicrobial peptide, that, similar to ALPS, consists of two α-helices connected by a flexible middle section.

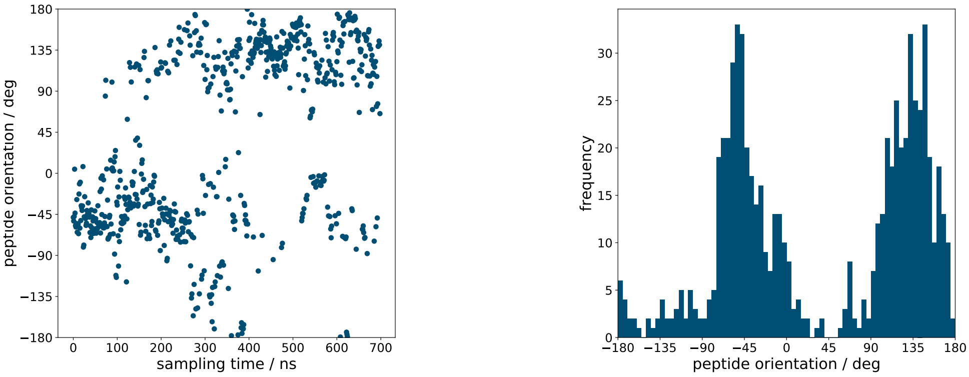
How to treat peptide orientation, i.e., to intregrate out or suppress this degree of freedom, or even resolve a 2d free energy with curvature K and orientation θ as reaction coordinates, is problem specific and depends mostly on the peptide size, structure, and internal degrees of freedom.

2.A.5 Hamiltonian replica exchange

To corroborate that there are only two preferred orientations of ALPS at $s = 0.5$, we employed Hamiltonian replica exchange (HREX) in combination with our otherwise unchanged umbrella sampling setup for one window only. Enhanced sampling enables us to discern whether infrequent visiting of states with $\theta \approx +45°$ and $\theta \approx -135°$ in the simulations is either caused by poor sampling or the existence of a free energy barrier against rotation.

The data shown in Fig. 2.11 support the finding that there is a symmetry between $\theta \approx -45°$ and $\theta \approx +135°$ and that $\theta \approx +45°$ and $\theta \approx -135°$ are indeed energetically unfavorable in comparison.

We followed an approach similar to Refs. [69, 70], where the system is divided in a *hot* and *cold* part. In the hot part, the force-field parameters are scaled down, such that interactions inside the hot region have an effective temperature of T/λ, with the scaling parameter λ. For interactions between hot and cold regions the effective temperature is $T/\sqrt{\lambda}$. Inside the cold region interactions are unchanged. In our system the peptide was defined as the hot region. We used 6 replicas and a geometric distribution of lambdas between 1 and 0.48. The simulations were 750 ns long, with the first 50 ns for equilibration. Simulations were performed using GROMACS 2019.6[72] patched with PLUMED 2.6.2[32, 73]. Otherwise, simulation parameters were set as described in the main paper.



(a) ALPS orientation on the tangent plane for the replica with $\lambda = 1$.

(b) Probability distribution of ALPS orientation on the tangent plane

Figure 2.11: Enhanced sampling of ALPS angular orientation with HREX.

Extension of our umbrella sampling protocol for buckled membranes with HREX is straightforward and recommended, especially for strongly curved regions, or where some kind of energy barrier is expected. Furthermore, regular free energy reconstruction methods such as UA and WHAM can be employed, since HREX does not alter the statistics of the unscaled system ($\lambda = 1$) [69].

2.A.6 Curvature and packing defects

An in-house modified version of PackMem [74] was used to calculate and visualize the lipid packing defects. To this aim, the standard version was extended to include calculation of hydrophobic defects with respect to the normal vector on the tangent plane of buckled membranes.

2.A.7 Change of variables of a potential of mean force

The PMF along a reaction coordinate ξ is defined from the probability density function $P(\xi)$[75],

$$F(\xi) = -k_{\mathrm{B}}T \ln\left[\frac{P(\xi)}{P(\xi^*)}\right] + \text{const.} \tag{2.41}$$

where ξ^* is an arbitrary constant. $P(\xi^*)$ is absorbed into the other arbitrary constant in most texts, explicitly including it ensures proper normalization later. From (2.41) it is clear that changing the variable of $F(\xi)$ means changing the variable of $P(\xi)$. This is done by the standard change-of-variables formula of a probability density function that is given in many textbooks on probability theory

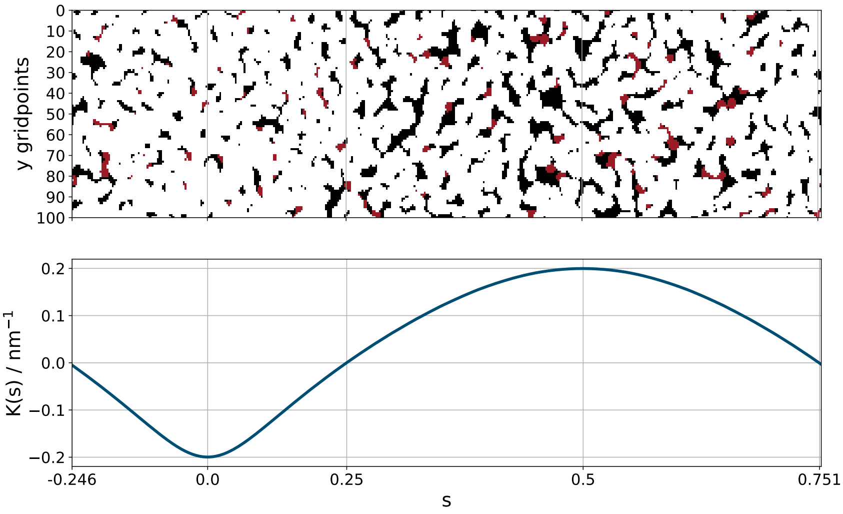


Figure 2.12: top: Lipid packing defects in the upper leaflet (black: deep defects, red: shallow). Unrolled topview. bottom: Curvature of the membrane midplane as a function of arc length parameter s.

and statistics, e.g., Held and Bové [76]. When $\gamma(\xi)$ is a one-to-one transformation and $\frac{\mathrm{d}\gamma(\xi)}{\mathrm{d}\xi}$ is nonzero, the transformed probability density is given by

$$P(\gamma(\xi)) = \frac{P(\xi)}{\left|\frac{\mathrm{d}\gamma(\xi)}{\mathrm{d}\xi}\right|}. \tag{2.42}$$

Which follows from the fact that the probability contained in $|P(\gamma(\xi))\mathrm{d}\gamma(\xi)|$ is the same as in $|P(\xi)\mathrm{d}\xi|$. Writing the PMF as a function of γ and plugging (2.42) into (2.41) yields

$$F(\gamma(\xi)) = -k_\mathrm{B}T \ln\left[\frac{P(\gamma(\xi))}{P(\gamma(\xi^*))}\right] + \text{const.} \tag{2.43a}$$

$$= -k_\mathrm{B}T \ln\left[\frac{P(\xi)}{P(\xi^*)} \frac{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|_{\xi=\xi^*}}{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|}\right] + \text{const.} \tag{2.43b}$$

$$= -k_\mathrm{B}T \ln\left[\frac{P(\xi)}{P(\xi^*)}\right] + k_\mathrm{B}T \ln\left[\frac{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|}{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|_{\xi=\xi^*}}\right] + \text{const.} \tag{2.43c}$$

$$= F(\xi) + k_\mathrm{B} T \ln \left[\frac{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|}{\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|_{\xi=\xi^*}} \right]. \tag{2.43d}$$

While the constant $\left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|_{\xi=\xi^*}$ is technically necessary to make the argument of the logarithm dimensionless, we absorb it into the arbitrary constant and write

$$F(\gamma) = F(\xi) + k_\mathrm{B} T \ln \left|\frac{\mathrm{d}\gamma}{\mathrm{d}\xi}\right|. \tag{2.44}$$

References

[1] Antonny, B. “Mechanisms of Membrane Curvature Sensing”. In: *Annu. Rev. Biochem.* vol. 80, no. 1 (2011), pp. 101–123.

[2] Baumgart, T. et al. “Thermodynamics and Mechanics of Membrane Curvature Generation and Sensing by Proteins and Lipids”. In: *Annu. Rev. Phys. Chem.* vol. 62, no. 1 (2011), pp. 483–506.

[3] Mondal, S. et al. “Membrane Driven Spatial Organization of GPCRs”. In: *Sci. Rep.* vol. 3, no. 1 (2013).

[4] Iversen, L. et al. “Membrane curvature bends the laws of physics and chemistry”. In: *Nat. Chem. Biol.* vol. 11, no. 11 (2015), pp. 822–825.

[5] Koldsø, H. and Sansom, M. S. P. “Organization and Dynamics of Receptor Proteins in a Plasma Membrane”. In: *J. Am. Chem. Soc.* vol. 137, no. 46 (2015), pp. 14694–14704.

[6] Rosholm, K. R. et al. “Membrane curvature regulates ligand-specific membrane sorting of GPCRs in living cells”. In: *Nat. Chem. Biol.* vol. 13, no. 7 (2017), pp. 724–729.

[7] Bhaskara, R. M. et al. “Curvature induction and membrane remodeling by FAM134B reticulon homology domain assist selective ER-phagy”. In: *Nat. Commun.* vol. 10, no. 1 (2019).

[8] Hofbauer, H. F. et al. “The molecular recognition of phosphatidic acid by an amphipathic helix in Opi1”. In: *J. Cell Biol.* vol. 217, no. 9 (2018), pp. 3109–3126.

[9] Pranke, I. M. et al. “α-Synuclein and ALPS motifs are membrane curvature sensors whose contrasting chemistry mediates selective vesicle binding”. In: *J. Cell Biol.* vol. 194, no. 1 (2011), pp. 89–103.

[10] Drin, G. et al. “A general amphipathic α-helical motif for sensing membrane curvature”. In: *Nat. Struct. Mol. Biol.* vol. 14, no. 2 (2007), pp. 138–146.

[11] Hatzakis, N. S. et al. “How curved membranes recruit amphipathic helices and protein anchoring motifs”. In: *Nat. Chem. Biol.* vol. 5, no. 11 (2009), pp. 835–841.

[12] Cui, H., Lyman, E., and Voth, G. A. “Mechanism of Membrane Curvature Sensing by Amphipathic Helix Containing Proteins”. In: *Biophys. J.* vol. 100, no. 5 (2011), pp. 1271–1279.

[13] Nepal, B., Sepehri, A., and Lazaridis, T. “Mechanisms of negative membrane curvature sensing and generation by ESCRT III subunit Snf7”. In: *Protein Sci.* vol. 29, no. 6 (2020), pp. 1473–1485.

[14] Cannon, K. S. et al. “An amphipathic helix enables septins to sense micrometer-scale membrane curvature”. In: *J. Cell Biol.* vol. 218, no. 4 (2019), pp. 1128–1137.

[15] Gallea, J. I. et al. “Amyloid oligomerization of the Parkinson’s disease related protein α-synuclein impacts on its curvature-membrane sensitivity”. In: *J. Neurochem.* vol. 147, no. 4 (2018), pp. 541–556.

[16] Peter, B. J. “BAR Domains as Sensors of Membrane Curvature: The Amphiphysin BAR Structure”. In: *Science* vol. 303, no. 5657 (2004), pp. 495–499.

[17] Hu, M., Diggins, P., and Deserno, M. “Determining the bending modulus of a lipid membrane by simulating buckling”. In: *J. Chem. Phys.* vol. 138, no. 21 (2013), p. 214110. eprint: https://doi.org/10.1063/1.4808077.

[18] Gómez-Llobregat, J., Elías-Wolff, F., and Lindén, M. “Anisotropic Membrane Curvature Sensing by Amphipathic Peptides”. In: *Biophys. J.* vol. 110, no. 1 (2016), pp. 197–204.

[19] Elías-Wolff, F. et al. “Computing Curvature Sensitivity of Biomolecules in Membranes by Simulated Buckling”. In: *J. Chem. Theory Comput.* vol. 14, no. 3 (2018), pp. 1643–1655.

[20] Elías-Wolff, F. et al. “Curvature sensing by cardiolipin in simulated buckled membranes”. In: *Soft Matter* vol. 15, no. 4 (2019), pp. 792–802.

[21] Martyna, A. et al. “Curvature Sensing by a Viral Scission Protein”. In: *Biochemistry* vol. 55, no. 25 (2016), pp. 3493–3496.

[22] Heinrich, M. C. et al. “Quantifying Membrane Curvature Generation of Drosophila Amphiphysin N-BAR Domains”. In: *J. Phys. Chem. Lett.* vol. 1, no. 23 (2010), pp. 3401–3406.

[23] Simunovic, M. et al. “Physical basis of some membrane shaping mechanisms”. In: *Phil. Trans. R. Soc. A* vol. 374, no. 2072 (2016), p. 20160034.

[24] Helfrich, W. “Elastic Properties of Lipid Bilayers: Theory and Possible Experiments”. In: *Z. Naturforsch. C* vol. 28, no. 11-12 (1973), pp. 693–703.

[25] Wassenaar, T. A. et al. “Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models”. In: *J. Chem. Theory Comput.* vol. 10, no. 2 (2014), pp. 676–690.

[26] Nguyen, H., Case, D. A., and Rose, A. S. “NGLview–interactive molecular graphics for Jupyter notebooks”. In: *Bioinformatics* vol. 34, no. 7 (2017). Ed. by Valencia, A., pp. 1241–1242.

[27] Gautier, R. et al. “HELIQUEST: a web server to screen sequences with specific -helical properties”. In: *Bioinformatics* vol. 24, no. 18 (2008), pp. 2101–2102.

[28] Kästner, J. “Umbrella sampling”. In: *WIREs Comput Mol Sci* vol. 1, no. 6 (2011), pp. 932–942.

[29] Torrie, G. and Valleau, J. “Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling”. In: *J. Comput. Phys.* vol. 23, no. 2 (1977), pp. 187–199.

[30] Kumar, S. et al. “The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method”. In: *J. Comput. Chem.* vol. 13, no. 8 (1992), pp. 1011–1021.

[31] Kästner, J. and Thiel, W. “Bridging the gap between thermodynamic integration and umbrella sampling provides a novel analysis method: “Umbrella integration””. In: *J. Chem. Phys.* vol. 123, no. 14 (2005), p. 144104.

[32] Tribello, G. A. et al. “PLUMED 2: New feathers for an old bird”. In: *Comput. Phys. Commun.* vol. 185, no. 2 (2014), pp. 604–613.

[33] Song, H. D. and Zhu, F. “Finite Temperature String Method with Umbrella Sampling: Application on a Side Chain Flipping in Mhp1 Transporter”. In: *J. Phys. Chem. B* vol. 121, no. 15 (2016), pp. 3376–3386.

[34] Otter, W. K. den. “Free energies of stable and metastable pores in lipid membranes under tension”. In: *J. Chem. Phys.* vol. 131, no. 20 (2009), p. 205101.

[35] Farouki, R. and Neff, C. “Analytic properties of plane offset curves”. In: *Comput. Aided Geom. Des.* vol. 7, no. 1-4 (1990), pp. 83–99.

[36] Kästner, J. and Thiel, W. “Analysis of the statistical error in umbrella sampling simulations by umbrella integration”. In: *J. Chem. Phys.* vol. 124, no. 23 (2006), p. 234106.

[37] Schiferl, S. K. and Wallace, D. C. “Statistical errors in molecular dynamics averages”. In: *J. Chem. Phys.* vol. 83, no. 10 (1985), pp. 5203–5209.

[38] Mann, H. B. “Nonparametric Tests Against Trend”. In: *Econometrica* vol. 13, no. 3 (1945), p. 245.

[39] D’Agostino, R. B. “An omnibus test of normality for moderate and large size samples”. In: *Biometrika* vol. 58, no. 2 (1971), pp. 341–348.

[40] Neumann, J. von. “Distribution of the Ratio of the Mean Square Successive Difference to the Variance”. In: *Ann. Math. Stat.* vol. 12, no. 4 (1941), pp. 367–395.

[41] Wassenaar, T. A. et al. “Computational Lipidomics with insane: A Versatile Tool for Generating Custom Membranes for Molecular Simulations”. In: *J. Chem. Theory Comput.* vol. 11, no. 5 (2015), pp. 2144–2155.

[42] Boyd, K. J., Alder, N. N., and May, E. R. “Buckling Under Pressure: Curvature-Based Lipid Segregation and Stability Modulation in Cardiolipin-Containing Bilayers”. In: *Langmuir* vol. 33, no. 27 (2017), pp. 6937–6946.

[43] Bussi, G., Donadio, D., and Parrinello, M. “Canonical sampling through velocity rescaling”. In: *J. Chem. Phys.* vol. 126, no. 1 (2007), p. 014101.

[44] Abraham, M. J. et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”. In: *SoftwareX* vol. 1-2 (2015), pp. 19–25.

[45] Lindahl et al. *GROMACS 2019.3 Source code*. en. 2019.

[46] Lindahl et al. *GROMACS 2020.3 Source code*. 2020.

[47] Marrink, S. J. et al. “The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations”. In: *J. Phys. Chem. B* vol. 111, no. 27 (2007), pp. 7812–7824.

[48] Monticelli, L. et al. “The MARTINI Coarse-Grained Force Field: Extension to Proteins”. In: *J. Chem. Theory Comput.* vol. 4, no. 5 (2008), pp. 819–834.

[49] Jong, D. H. de et al. “Improved Parameters for the Martini Coarse-Grained Protein Force Field”. In: *J. Chem. Theory Comput.* vol. 9, no. 1 (2012), pp. 687–697.

[50] Michaud-Agrawal, N. et al. "MDAnalysis: A toolkit for the analysis of molecular dynamics simulations". In: *J. Comput. Chem.* vol. 32, no. 10 (2011), pp. 2319–2327.

[51] Gowers, R. et al. "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations". In: *Proceedings of the 15th Python in Science Conference*. SciPy, 2016.

[52] Kapla, J. and Lindén, M. *Mxdrfile: read and write Gromacs trajectories with Matlab*. 2018. arXiv: `1811.03012v1 [physics.comp-ph]`.

[53] Hilten, N. van, Stroh, K. S., and Risselada, H. J. "Membrane Thinning Induces Sorting of Lipids and the Amphipathic Lipid Packing Sensor (ALPS) Protein Motif". In: *Front. Physiol.* vol. 11 (2020).

[54] González-Rubio, P. et al. "Amphipathic-Lipid-Packing-Sensor interactions with lipids assessed by atomistic molecular dynamics". In: *Biochimic. Biophys. Acta - Biomembr.* vol. 1808, no. 9 (2011), pp. 2119–2127.

[55] Kabsch, W. and Sander, C. "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features". In: *Biopolymers* vol. 22, no. 12 (1983), pp. 2577–2637.

[56] Touw, W. G. et al. "A series of PDB-related databanks for everyday needs". In: *Nucleic Acids Res.* vol. 43, no. D1 (2014), pp. D364–D368.

[57] González-Rubio, P. et al. "Amphipathic-Lipid-Packing-Sensor interactions with lipids assessed by atomistic molecular dynamics". In: *Biochim. Biophys. Acta - Biomembr.* vol. 1808, no. 9 (2011), pp. 2119–2127.

[58] Rao, J. N. et al. "A Combinatorial NMR and EPR Approach for Evaluating the Structural Ensemble of Partially Folded Proteins". In: *J. Am. Chem. Soc.* vol. 132, no. 25 (2010), pp. 8657–8668.

[59] Daum, G. "Lipids of mitochondria". In: *Biochim. Biophys. Acta, Rev. Biomembr.* vol. 822, no. 1 (1985), pp. 1–42.

[60] Varkey, J. et al. "Membrane Curvature Induction and Tubulation Are Common Features of Synucleins and Apolipoproteins". In: *J. Biol. Chem.* vol. 285, no. 42 (2010), pp. 32486–32493.

[61] Sorre, B. et al. "Nature of curvature coupling of amphiphysin with membranes depends on its bound density". In: *Proc. Natl. Acad. Sci. USA* vol. 109, no. 1 (2011), pp. 173–178.

[62] Zhu, C., Das, S. L., and Baumgart, T. "Nonlinear Sorting, Curvature Generation, and Crowding of Endophilin N-BAR on Tubular Membranes". In: *Biophys. J.* vol. 102, no. 8 (2012), pp. 1837–1845.

[63] Shi, Z. and Baumgart, T. “Membrane tension and peripheral protein density mediate membrane shape transitions”. In: *Nat. Commun.* vol. 6, no. 1 (2015).

[64] Chen, Z., Shi, Z., and Baumgart, T. “Regulation of Membrane-Shape Transitions Induced by I-BAR Domains”. In: *Biophys. J.* vol. 109, no. 2 (2015), pp. 298–307.

[65] Bubnis, G., Risselada, H. J., and Grubmüller, H. “Exploiting Lipid Permutation Symmetry to Compute Membrane Remodeling Free Energies”. In: *Phys. Rev. Lett.* vol. 117, no. 18 (2016).

[66] Lipowsky, R. “Spontaneous tubulation of membranes and vesicles reveals membrane tension generated by spontaneous curvature”. In: *Faraday Discuss.* vol. 161 (0 2013), pp. 305–331.

[67] Różycki, B. and Lipowsky, R. “Spontaneous curvature of bilayer membranes from molecular simulations: Asymmetric lipid densities and asymmetric adsorption”. In: *J. Chem. Phys.* vol. 142, no. 5 (2015), p. 054101.

[68] Nepal, B., Leveritt, J., and Lazaridis, T. “Membrane Curvature Sensing by Amphipathic Helices: Insights from Implicit Membrane Modeling”. In: *Biophys. J.* vol. 114, no. 9 (2018), pp. 2128–2141.

[69] Bussi, G. “Hamiltonian replica exchange in GROMACS: a flexible implementation”. In: *Mol. Phys.* vol. 112, no. 3-4 (2013), pp. 379–384.

[70] Wang, L., Friesner, R. A., and Berne, B. J. “Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2)”. In: *J. Phys. Chem. B* vol. 115, no. 30 (2011), pp. 9431–9438.

[71] Yesylevskyy, S. O., Rivel, T., and Ramseyer, C. “The influence of curvature on the properties of the plasma membrane. Insights from atomistic molecular dynamics simulations”. In: *Sci. Rep.* vol. 7, no. 1 (2017).

[72] Lindahl et al. *GROMACS 2019.6 Source code*. en. 2020.

[73] “Promoting transparency and reproducibility in enhanced molecular simulations”. In: *Nat. Methods* vol. 16, no. 8 (2019), pp. 670–673.

[74] Gautier, R. et al. “PackMem: A Versatile Tool to Compute and Visualize Interfacial Packing Defects in Lipid Bilayers”. In: *Biophys. J.* vol. 115, no. 3 (2018), pp. 436–444.

[75] Roux, B. “The calculation of the potential of mean force using computer simulations”. In: *Comput. Phys. Commun.* vol. 91, no. 1-3 (1995), pp. 275–282.

[76] Held, L. and Bové, D. S. *Applied Statistical Inference*. Springer Berlin Heidelberg, 2014.

Chapter 3

CGCompiler: Automated coarse-grained molecule parameterization via noise-resistant mixed-variable optimization

Kai Steffen Stroh, Paulo C. T. Souza, Luca Monticelli, Herre Jelger Risselada

Reprinted with permission from
Journal of Chemical Theory and Computation,
2023, volume 19, issue 22, pp. 8384-8400.
DOI: 10.1021/acs.jctc.3c00637.

https://pubs.acs.org/articlesonrequest/AOR-HRZSDHKTNEUDWIQB2BDW

Contents

3.1 Introduction

Atomically detailed molecular dynamics (MD) simulations provide great insights into the structure and dynamics of biomolecular and other soft matter systems, but larger time- and length scales often require a coarse-grained (CG) description. In coarse-graining a group of atoms is mapped into one bead or supra-atom. Coarse-grained descriptions achieve computational efficiency by reducing degrees of freedom while preserving relevant aspects. This not only allows for bridging larger time and length scales but also enhances our understanding of the fundamental physics underlying molecular processes within biological cells. For example, it can enable fundamental insights into phenomena like the self-organization of lipid membranes and the formation of characteristic thermodynamic phases, including liquid-ordered, liquid-disordered, and gel phases [1–3]. Systematic coarse-graining approaches such as inverse Boltzmann and inverse Monte-Carlo approaches [4, 5] as well as force-matching approaches [6, 7] parameterize coarse-grained force-fields by reproducing the structural part of the partition function of the fine-grained system by either matching relevant radial distribution functions or (combined) forces within the fine-grained system. However, because the partition function only describes a single thermodynamic state point at equilibrium, i.e., a unique combination of pressure & temperature values, systematically parameterized ’bottom-up’ coarse-grained force-fields are not suited to describe phase transitions over a wider temperature range. Phase-transitions or phase-diagrams can, however, be optimally modeled using coarse-grained force-fields based on the alternative Statistical Associating Fluid Theory (SAFT) parameterization approach, which uses a scaled Lennard-Jones interaction potential whose functional form (the exponent) is uniquely adapted for each interaction type [8, 9]. However, the main practical problem of all of these coarse-grained force-fields is their lack of chemical transferability, i.e. inclusion of a new molecule (interaction type) within the system would require reparameterization of all the existing interaction parameters.

The Martini coarse-grained force-field [10, 11] is a building block Force-Field (FF), i.e., common chemical groups are parameterized as basic building blocks, which can be combined to build up any existing molecule. These basic building blocks of Martini, the beads, are parameterized top-down and reproduce the thermodynamic properties of the chemical groups they model, such as partitioning free energies in liquid-liquid systems, while complete molecules are parameterized with a combination of top-down (experimental data) and bottom-up (atomistic simulation). Such a parameterization enables the qualitative simulation of phase transitions as well as phase segregation in lipid membranes while simultaneously conserving molecular compatibility (transferability) by describing all non-bonded interactions with the same 12-6 Lennard-Jones potential form. However, a major drawback compared to other systematic coarse-grained approaches is that parameterization of molecules in Martini can be highly complex and often involves the selection and fine tuning of a large number of parameters (e.g., bead types and bond lengths) to optimally match multiple relevant targets simultaneously. A task that is time consuming when done by human labor. Additionally, it is not always obvious which parameters have to be changed in what manner to enhance a certain behavior, particularly when cooperative processes are involved. While the choice of individual bead types can be made using chemical intuition, still a sizable subset of combined possibilities exists. Importantly, parameterization of bonded and non-bonded parameters should be optimally performed simultaneously since bonded and non-bonded interactions are not independent – they are directly influencing each other via the density of interactions [12, 13]. Recent versions of the Martini force-fields such as Martini 3 rebalanced the density of interactions by introducing an even larger number of possible interaction types, thereby rendering the parameterization of molecules often a non-tractable problem to common users. Automation of coarse-graining is thus critical, especially when constructing large databases of molecules. Automation offers a solution to address the challenge of force-field development, which typically involves collaboration among multiple researchers working on interdependent parameters. The automation approach therefore facilitates collaborations by allowing researchers to focus on selecting a set of relevant objectives and assigning importance or weights to each objective. These objectives, along with their individual weights, define the force-field's philosophy. Furthermore, automation empowers collaborations to prioritize two key aspects: the generation and provision of reference data for the objectives at hand, and the design of analysis tools to quantitatively assess how each objective is addressed within the automation pipeline. By automating the parameterization process, collaborators can allocate their efforts towards obtaining high-quality reference data that accurately represents the desired objectives. Simultaneously, they can focus on developing comprehensive analysis tools that enable thorough quantitative evaluation, ensuring the effectiveness of the automation pipeline in achieving the defined objectives. This collaborative approach maximizes the efficiency, reliability, and reproducibility of the parameterization process while facilitating a deeper understanding of the force field's performance.

Earlier works on automated parameterization for building block FFs focused on optimizing bonded interactions only [14–16]. For example, a method such as PyCGTOOL generates coarse-grained model parameters from atomistic simulation trajectories using a user-provided mapping. However, it does not perform parameter optimization, instead equilibrium values and force constants, are generated by Boltzmann inversion [14]. No other targets are used. The SwarmCG method performs parameter optimization with traditional PSO and targets only bond length and angle distributions, as well as bilayer dimensions [16]. The melting temperature is only used in validation after optimization. Non-bonded parameters are not being optimized, although a previous SwarmCG implementation [17] could also perform optimization of continuous non-bonded parameters. No bead assignment is proposed, which is problematic for molecule parameterization in building block FFs, as explained further down. Other approaches that addressed both the automation of mapping as well as the parametrization of bonded and non-bonded parameters solely focused on small molecules, and rather provide an initial guess than an optimized parameterization [18, 19]. In Auto-Martini bead type selection is done via ALOGPS [20, 21] partitioning prediction of fragments [18]. Bonded parameters use generic values, without any optimization. The approach from Potter et al. is similar to Auto-Martini, but features an improved mapping scheme, and non-bonded interactions are derived in a similar fashion, bond lengths are taken from relaxed atomistic structures, and the force constants use generic values [19]. We note that fast methods such as Auto-Martini and the method from Potter et al. could be used as a complementary approach to CGCompiler by providing an initial mapping as well as an initial non-bonded/bonded parameter guess for CGCompiler. Automation schemes exist also for systematic coarse-graining approaches [22, 23].

Particle swarm optimization (PSO) is a powerful computational method used to optimize problems by iteratively improving candidate solutions based on a defined objective function. Compared to evolutionary optimization methods like genetic algorithms, PSO offers advantages in efficiently finding global optima within high-dimensional continuous spaces due to its vectorial search direction. PSO has been successfully employed in various coarse-grained (CG) parameterization tasks, as demonstrated in previous studies [15–17, 24, 25].

PSO is primarily designed for continuous variables, making it well-suited for optimizing structure-based coarse-grained (CG) models where bonded and non-bonded parameters can be chosen from a continuum of values. However, in building block models like Martini, the non-bonded parameters are predefined and discrete, representing different interaction levels. Consequently, the parameterization of molecules in a building block CG force field becomes a mixed-variable optimization problem.

When using PSO for parameterization in building block models, a transformation from the continuous space to the discrete space of force field parameters is necessary. This transformation introduces cumulative rounding errors, which can potentially affect the quality of the parameterization, especially in larger molecules. Therefore, additional evaluation and reparameterization steps are often required to ensure the optimal performance of the force field.

It is crucial to parameterize both bonded and non-bonded interactions simultaneously since they are not independent and their optimization should be performed in a coordinated manner [13]. By considering their interplay during the parameterization process, the resulting force field can better capture the complex behavior of molecules in the system.

To address the limitations of existing PSO approaches, we employ a mixed-variable PSO scheme (mv-PSO) for parameterization. This approach allows for the simultaneous optimization of both discrete parameters (representing non-bonded interactions) and continuous parameters (representing bonded interactions), enhancing the accuracy and reliability of the parameterization process.

Furthermore, due to the chaotic nature of MD simulations, observables measured in MD simulations are subject to noise. Since standard PSO was designed for deterministic objective functions, straightforward application to noisy optimization problems is error prone, because the algorithm can no longer correctly identify global and personal best solutions when noise levels are similar to differences between objective function values [26]. Noise-mitigation strategies are particularly important when utilizing thermodynamic data as targets, as these are notoriously expensive to estimate accurately in MD simulations, even when employing CG models. Particularly problematic is the targeting of phase transition temperatures, which involve a first order phase transition and are thus subject to nucleation and concomitant hysteresis.

In this paper, we pioneer the application of mixed-variable particle swarm optimization in automated parameterization of molecules within the Martini 3 coarse-grained force-field by matching both structural (e.g., RDFs) as well as thermodynamic data (phase-transition temperatures). The important advantage of this approach is that both bonded- and non-bonded interactions are simultaneously optimized while conserving the search efficiency of vector guided particle swarm methods over other metaheuristic search methods such as genetic algorithms. In addition, we explore noise-mitigation strategies in matching the phase transition temperatures, where nucleation and concomitant hysteresis introduces a dominant noise term within the objective function. To the best of our knowledge, the impact of noisy objective function values has not been previously addressed in the context of applying PSO for CG parameterization. The manuscript is structured in the following way: Section 3.2 describes the mixed-variable PSO algorithm and parameterization procedure. As an example, we parameterized the linker region of sphingolipids, a biological highly relevant class of lipid molecules, that constitutes approximately 30 mol% of the plasma membrane lipids [27], but has not been updated for Martini 3, yet. Details of the simulated molecules, systems and observables are given in Section 3.3. Results are presented in Section 3.4, followed by conclusions in Section 3.5.

3.2 CG molecule parameterization via mixed-variable particle swarm optimization

With CGCompiler we present a Python package that streamlines CG molecule parameterization. It employs mixed-variable particle swarm optimization to simultaneously optimize categorical (beadtype) and continuous (bonds, angles, dihedrals, ...) variables. Therefore, CGCompiler is particularly well suited for, but not limited to, parameterization tasks in CG FFs that follow a building block approach. To enable the application of the building block approach also to larger molecular fragments, consisting of more than one CG bead, the method allows for optimization of shared building blocks in different molecules, e.g. the headgroup, linker, or tails of lipids.

Molecule parameterization in Martini 3 follows three steps: i) Choice of mapping and bead sizes ii) Assignment of chemical bead types iii) Choice of bonded terms and assignment of bonded parameters [11]. While a mapping from atomistic to CG model and the set of of bonded terms have to be predefined, the here-presented algorithm optimizes bead size, chemical bead type and bonded parameters simultaneously.

The parameterization workflow is shown in Figure 3.1. For a given parameterization task, the user provides or generates the target data, and creates a set of CG training systems, that allows measurement of the target observables. In the initial iteration, the optimization algorithm generates a number N_p, i.e., the swarm size, of candidate solutions with random FF parameters, and runs MD simulations for each candidate solution and each training system. Candidate solutions are then scored by how well the parameterization targets are reproduced. By utilizing the swarm's knowledge of the fitness landscape, candidate solutions are updated, and a new cycle of MD simulations, analyses, and fitness evaluations, starts. This is repeated until a termination criterion is fulfilled. Due to noise in the objective function evaluation, the selection of the true best parameters can only be done with a certain probability. Therefore, the set of the best, statistically equal candidate solutions undergoes a screen-to-the-best procedure, which either provides one solution that is significantly better than the rest, or reduces the

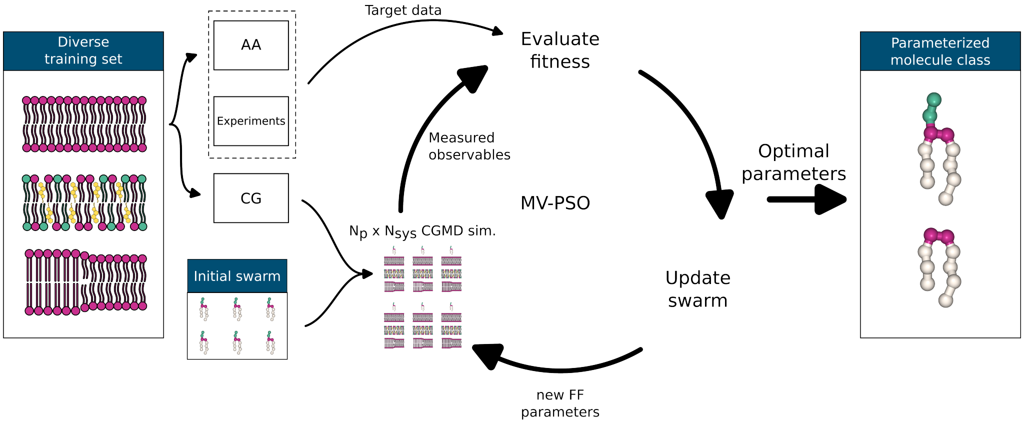


Figure 3.1: Parameterization workflow. i) A set of training systems from which the target properties can be extracted. ii) Target data is acquired from atomistic simulations and experiments. iii) An initial swarm is generated with FF parameters randomly selected from a predefined range of feasible parameters. iv) All candidate solutions are simulated in all training systems, the target observables are measured and compared to the target data, i.e., the fitness of the candidate solutions is estimated. New candidate solutions are generate by utilizing the swarm's knowledge of the fitness landscape. v) Step iv is repeated until a termination criterion is fulfilled. vi) A screen-to-the best procedure yields the optimized set of FF parameters.

field of viable candidate solutions further, on which more expensive evaluation simulations would be performed.

3.2.1 Mixed-variable particle swarm optimization

In the original PSO algorithm for continuous optimization problems in a D-dimensional parameter space, particle i has a position vector $X_i = (x_i^1, ..., x_i^D)$ and a velocity $V_i = (v_i^1, ..., v_i^D)$ [28]. At each iteration t the velocity and position are updated by

$$\begin{aligned} V_i(t+1) = w * V_i(t) & \\ + c_1 \boldsymbol{r}_1(\text{pbest}_i(t) - X_i(t)) & \qquad (3.1)\\ + c_2 \boldsymbol{r}_2(\text{gbest}(t) - X_i(t)) & \end{aligned}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (3.2)$$

Where $\text{pbest}_i(t)$ is the personal best position of particle i and $\text{gbest}(t)$ is the best position found by the whole swarm. w is an inertia weight, which balances global vs. local search. The coefficients c_1 and c_2 are balancing personal vs.

social experience. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of random numbers. In the mv-PSO algorithm, that is utilized in our work, the position vector of a particle takes a hybrid form, where Z dimensions encode continuous variables and V dimensions encode categorical variables [29].

$$X_i = (\underbrace{x_i^1, x_i^2, ..., x_i^Z}_{\text{continuous}}, \underbrace{x_i^{Z+1}, x_i^{Z+2}, ..., x_i^{Z+V}}_{\text{categorical}}) \tag{3.3}$$

The continuous and categorical parts of the position vector are updated separately.

3.2.1.1 Continuous reproduction method

In classical PSO the swarm can get trapped in local optima and therefore prematurely converge [29]. To promote diversity while maintaining good convergence efficiency Wang et al. proposed an altered continuous reproduction scheme, where particle i learns from the best position of a randomly selected particle [29]. In order to guide the swarm towards improved solutions, the pool of pbest to choose from, only consists of solutions whose fitness is superior to $\text{pbest}_i(t)$.

$$V_i(t+1) = w \cdot V_i(t) + c \cdot \boldsymbol{r} \cdot (\text{pbest}_r(t) - X_i(t)) \tag{3.4}$$

Algorithm 1 Continuous reproduction method

```
Input: sorted swarm, particle $i$, parameter $w_i$
for $j = 1..Z$ do
    Randomly choose $r$, $i \le r \le N$
    $v_i^j(t+1) = w_i \cdot v_i^j(t) + c \cdot r \cdot (pbest_r^j - x_i^j)$
    $x_i^j(t+1) = x_i^j(t) + v_i^j(t+1)$
end for
return $(x_i^1, x_i^2, ..., x_i^Z)$
```

3.2.1.2 Categorical reproduction method

Values of categorical variables are assigned according to a probability. Initial probabilities are given by

$$Prob_{j,n}(0) = \frac{1}{n_j} \tag{3.5}$$

where n_j is the number of available values for the jth variable. To leverage the swarm's knowledge of good solutions, only the superior half of the sorted swarm is utilized in updating the probabilities of available categorical values. To avoid premature extinction of available values, a lower limit is assigned for $Prob_{j,n}$. If

$Prob_{j,n}$ falls below that lower limit, $Prob_{j,n}$ is set to that threshold value, and all probabilities are renormalized such that $\sum_n Prob_{j,n} = 1$. The categorical update method is shown in Algorithm 2.

Algorithm 2 Categorical reproduction method

```
Input: sorted swarm, particle $i$, parameter $\alpha_i$
for $j = 1..V$ do
    for each available value $n$, $n = 1$ to $n_j$ do $Count_{j,n} = 0$
        for each personal best $pbest_i$, $i = N/2$ to $N$ do
            if $pbest_{i,j} == Values_{j,n}$ then
                $Count_{j,n} + = 1$
            end if
        end for
        $Prob_{j,n}(t+1) = \alpha_i \cdot Prob_{j,n}(t) + (1 - \alpha_i) \cdot \frac{Count_{j,n}}{N/2}$
    end for
end for
for $j = 1..V$ do
    Assign an available value to $x_i^{Z+j}$ according $Prob_j$
end for
return $(x_i^{Z+1}, x_i^{Z+2}, ..., x_i^{Z+V})$
```

3.2.1.3 Cost function

Molecule parameterization is typically a multiobjective optimization problem (MOP). A simple way to scalarize an MOP is by linear weighting. The scalarized optimization problem is solved by minimizing the cost, which is given by

$$\text{cost} = \sum_o w_o f_o(\boldsymbol{x}) \tag{3.6}$$

Where w_o is an objective weight, f_o the objective cost function, and $\boldsymbol{x}$ the parameter vector. The objective weights can be used to balance the importance of the utilized parameterization targets. The weights are set by the user. Setting weights might require some intuition about the parameterized molecule, quality of target data, etc.

Each objective can have a different objective cost function f_o. New objective cost functions can be added by the user easily. In its present form, the parameterization algorithm uses two distinct objective cost functions. For *single valued observables*, such as area per lipid, membrane thickness, melting temperature, solvent accessible surface area (SASA) the objective cost function is defined as

$$f_o(\boldsymbol{x}) = \frac{1}{\sum_s^{N_s} w_{o,s}} \left(\sum_s^{N_s} w_{o,s} \frac{1}{N_{\text{types},s}} \sum_t^{N_{\text{types},s}} \max(0, SAE(y_{s,t}(\boldsymbol{x}), \hat{y}_{s,t}) - E_{o,s}^{\text{tol}}) \right). \quad (3.7)$$

$y_s(\boldsymbol{x})$ is the observed value, given the FF parameters $\boldsymbol{x}$. $\hat{y}_s$ is the target value. N_s is the number of training systems that is used for the current parameterization objective. N_{types} is the number of bond or angle types being parameterized. The deviation from the target is calculated by the scaled absolute error $SAE(y, \hat{y}) = \left|\frac{\hat{y}-y}{\hat{y}}\right|$. With the error tolerance $E_{o,s}^{tol}$, uncertainties in target data can be accounted for. Each training system has an additional weight $w_{o,s}$, which can be used in case of differences in target data quality or similar cases. Generally these are set to 1.

For observables that are given in the form of *distributions*, such as bond lengths, angles, or radial distribution functions (RDF), the objective cost function is given by:

$$f_o(\boldsymbol{x}) = \frac{1}{\sum_s w_{o,s}} \left(\sum_s w_{o,s} \frac{1}{N_{\text{types},s}} \sum_t^{N_{\text{types},s}} EMD(\phi(\boldsymbol{x}_{s,t}), \hat{\phi}_{s,t}) \right) \quad (3.8)$$

Where $\phi(\boldsymbol{x})$ is the observed distribution, given the FF parameters $\boldsymbol{x}$. $\hat{\phi}$ is the target distribution. The earth mover's distance $EMD(\phi(\boldsymbol{x}_{s,t}), \hat{\phi}_{s,t})$ is a measure of the distance between the two distributions [30].

3.2.2 Noise mitigation strategies for PSO

PSO was designed for deterministic objective functions. Due to the chaotic nature of MD simulations hereby measured observables are subject to noise. With noise in objective functions, selection of the true best solutions is not guaranteed. Since solutions, that are identified as the best, attract the swarm toward regions of interest in parameter space, noise can misguide the swarm and therefore deteriorate PSO performance.

3.2.2.1 Resampling

Resampling is a widely applied strategy for noise mitigation within the objective function. Relatively simple resampling methods are *equal resampling* (PSO-ER), *extended equal resampling* PSO-EER, and *equal resampling* with allocation to top-N solutions PSO-ERN[31]. These simpler methods are regularly outperformed by state-of-the-art resampling methods, such as *optimal computing budget allocation* PSO-OCBA [32], but the quality of results depends on the specific optimization problem and noise levels[26, 31]. OCBA aims to maximize the probability of correctly selecting good solutions. This is done by first allocating a primary computational budget equally to all current solutions to estimate their cost means and variances. A secondary budget is then sequentially allocated to solutions with lower means and higher variances to improve the fitness estimations of potentially good solutions. For efficient secondary budget allocation at least 5 primary evaluations should be executed for mean and variance estimation [33]. This might make application of OCBA prohibitively expensive for regular CG molecule parameterization tasks. Based on the observation that most observables utilized in the multiobjective optimization of the sphingomyelin (SM) linker region have a low variance and only a few suffer from a larger variance (cf. Figure S5), we hypothesize that in the molecule parameterization task at hand, one primary objective function evaluation is sufficient to differentiate potentially good solutions from bad solutions, but to maximize the probability of correctly selecting the true best solution, the accuracy of the fitness estimates has to be increased. Therefore, we propose a somewhat pragmatic approach, that salvages the core idea of OCBA, i.e., allocate additional computational budgets to where it is the most useful (low mean and high variance). At each iteration, our resampling method involves one full objective function evaluation of the current solutions. The current solutions are then ranked by their fitness, and for the best N solutions only the observables that have significant variance are reevaluated.

3.2.2.2 Set of statistically equivalent solutions

Even with noise mitigation, at the end of an optimization run, there will be a number of solutions with very similar scores. While in a deterministic setting, the global best position is determined by

$$\boldsymbol{gbest} = \arg\min_{\boldsymbol{x}\in\mathcal{P}_t} f(\boldsymbol{x}), \tag{3.9}$$

where $\mathcal{P}_t$ is the set of all positions that have been visited by the swarm up to iteration t, with noise in the objective function no solution can be declared the best with 100% certainty [26]. With the *screen-to-the-best* procedure of Boesel et al. [34] a set of positions $\mathcal{P}_t^g \subseteq \mathcal{P}_t$ can be selected, such that the true global best solution $\boldsymbol{gbest}$ is contained in $\mathcal{P}_t^g$ with probability of at least $1 - \alpha$ (with $0 < \alpha < 1$) [26].

For solutions $i, j \in \mathcal{P}_t$, $\bar{f}_i$ and S_i^2 denote the sample mean and sample variance of objective function values. The elementary steps of the screen-to-the-best procedure are:

1. Compute W_{ij},

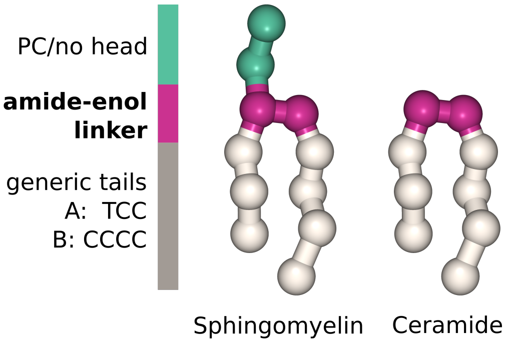


Figure 3.2: CG description of sphingomyelin and ceramide.

$$W_{ij} = \left(\frac{t_i S_i^2}{n_i} + \frac{t_j S_j^2}{n_j} \right)^{1/2}, \forall\, i \neq j \in \mathcal{P}_t \tag{3.10}$$

where $t_i = t_{(1-\alpha)^{1/|\mathcal{P}_t|-1}, n_i - 1}$ and $t_{\beta,\nu}$ is the β quantile of the t distribution with ν degrees of freedom

2. Set $\mathcal{P}_t^g = \{i : i \in \mathcal{P}_t, \bar{f}_i \leq \bar{f}_j + W_{ij}, \forall\, i \neq j \in \mathcal{P}_t\}$
3. Return $\mathcal{P}_t^g$

W_{ij} is the half-width of pooled t-confidence intervals on the difference between the scores of solutions i and j [26]. Therefore, the procedure entails a pair-wise comparison of solutions and determines if differences of the sample averaged scores are statistically significant [26].

3.3 Example application: Sphingolipid linker parameterization

As an example application of CGCompiler, we reparameterize the linker region of sphingomyelin in Martini 3 [11]. Fig. 3.2 depicts the CG models of two sphingolipids, sphingomyelin and ceramide. Except for the differing head group, the two CG models share the same parameters, following Martini's building block approach.

3.3.1 Simulation details

The Python package is based on evo-MD [35]. All simulations were performed with GROMACS 2020.4 and 2021.4 [36] and analyzed with in-house Python scripts that are utilizing MDAnalysis [37, 38], LiPyphilic [39], SciPy[40], and pyemd, which is a Python wrapper for Pele and Werman's EMD implementation [41, 42]. Visualization was done with NGLview [43].

3.3.1.1 Atomistic models

All atomistic models were simulated using the CHARMM36 [44–46] force field. Table 3.1 provides details about the atomistic target systems. Initial configurations of the membrane systems were generated with the CHARMM-GUI membrane builder [47–49]. Following energy minimization and equilibration, all systems were simulated with a 2 fs time step. Bonds of hydrogen atoms were constrained employing the LINCS algorithm [50]. Van der Waals forces were gradually switched off between 1.0 nm and 1.2 nm. The PME algorithm [51] was used for electrostatic interactions. Temperature coupling was done via the velocity rescale algorithm [52] with a coupling time $\tau_\mathrm{t} = 1.0$ ps. System pressures were held at 1 bar by using the Parinello-Rahman barostat [53] with a coupling time $\tau_\mathrm{p} = 5.0$ ps. Pressure coupling was applied isotropically for aqueous solutions and semi-isotropically for membrane systems.

system	lipids	# TIP3P	# NA	# CL	T / K	sim. time / ns
DPSM128 328K	128 SSM	5120	-	-	328.15	150
POPC SSM CHOL	100 POPC 100 SSM 100 CHL1	9000	18	18	321.15	300

Table 3.1: Atomistic target system details. In the naming scheme of the CHARMM FF, SSM and CHL1 denote sphingomyelin (18:0) and cholesterol, respectively.

3.3.1.2 Coarse-grained models

All coarse-grained models were simulated using the Martini 3 [11] force field. Beta version 14 of the Martini 3 cholesterol parameters was used [54, 55]. Initial configurations of membrane systems were generated with the Python script insane [56]. Details of the employed training systems are listed in Table 3.2. All systems were energy minimized and equilibrated with the current version of DPSM, that made the Martini 2 model of sphingomyelin compatible with Martini 3. During the particle swarm optimization each system was equilibrated with the candidate FF parameters in two stages, with time steps of 2 fs and 20 fs, respectively. For all coarse-grained production simulations a time step of 20 fs was used. Non-bonded interactions were cut off at 1.1 nm. For electrostatic interactions the reaction-field method was used with a dielectric constant of 15 and the reaction-field dielectric constant was set to infinity.

Temperature coupling was obtained via the velocity rescale algorithm [52] with a coupling time $\tau_\mathrm{t} = 1.0\,\mathrm{ps}$. System pressures were held at 1 bar by using the Parinello-Rahman barostat [53] with a coupling time $\tau_\mathrm{p} = 12.0\,\mathrm{ps}$. Pressure coupling was applied isotropically for aqueous solutions and semi-isotropically for membrane systems. In simulations for melting temperature estimation anisotropic pressure coupling was employed, using the Berendsen barostat [57] with a coupling time $\tau_\mathrm{p} = 4.0\,\mathrm{ps}$.

system	lipids	# W	# NA	# CL	T / K
DPSM128 328K	128 DPSM	1177	-	-	328.15
DPSM256 biphasic	256 DPSM half gel/half liquid	2300	26	26	286, 291, 296, 301, 303, 305, 307, 308, 309, 310, 311, 316, 321, 326
POPC SSM CHOL	96 POPC 96 DPSM 96 CHOL	2124	23	23	321.15

Table 3.2: Coarse-grained training system details.

3.4 Results

Our aim was the development of an automatization framework for molecule parameterization in building-block force fields. As an example we parameterized the sphingolipid linker region. Section 3.4.1 shows the results of the parameterization with CGCompiler using a simple noise-mitigation strategy. Since noise-mitigation strategies can only reduce the effects of noise when selecting the true best solution, the best statistically equivalent solutions generated during the mv-PSO run are subsequently screened-to-the-best, as described in Section 3.2.2.2.

3.4.1 Parameterization of the sphingolipid linker region

Table 3.3 shows the observables and their weights used in the parameterization. The swarm size was 64. Noise-mitigation was done by reevaluating the melting temperature of the 16 best candidate solutions of the current iteration 12 times, i.e., results were obtained with noise-mitigation setting mv-PSO-R16 (cf. Section 3.4.2). As T_m is the major contribution to cost variance, but the employed T_m estimation method is good for differentiating good from bad solutions, i.e., it has an accuracy of a few K. Other observables were only evaluated once, area per lipid (APL) fluctuations were the second largest cause of cost variance. For more details on noise-mitigation efficacy see Section 3.4.2.

All results shown include the complete set of the best statistically equivalent candidate solutions $\mathcal{P}^\mathrm{g}$ that remained after two rounds of the screen-to-the-best procedure (cf. Section 3.2.2.2). This set contains 18 candidate solutions.

observable	w_o	$w_\mathrm{o,\,DPSM128}$	$w_\mathrm{o,\,DPSM256}$	$w_\mathrm{o,\,POPC\ SSM\ CHOL}$
bond length dist.	1	1	0	1
angle dist.	100	1	0	1
d_HH	500	1	0	0.25
APL	1000	1	0	0.25
T_m	250	0	1	0
RDF COM DPSM-CHOL	1	0	0	1

Table 3.3: Weights of observables w_o and system specific observable weights $w_\mathrm{o,s}$ for optimization run 1.

3.4.1.1 Improved reproduction of membrane properties

Figure 3.3 shows thickness, average area per lipid and melting temperature of pure DPSM membranes for the set of statistically equal candidate solutions that remained after the second screen-to-the-best procedure performed after reevaluating the initial set 20 times. All new candidate solutions outperform the current DPSM model regarding thickness. The average area per lipid of the current model is closer to the target value, but most of the candidate solutions are within the tolerance of 1.5% deviation. In general, thickness and APL are inversely correlated, increasing one will always result in decreasing the other, therefore, with both values inside the tolerance, the new models represent a better balance of thickness and APL. It is important to note that in the comparison, SM(18:0) was used as the atomistic target. The current tail model of the Martini FF represents both SM(16:0) and SM(18:0). The CHARMM model for SM(16:0) exhibits a reduced thickness when compared to SM(18:0) [45]. It is therefore not unexpected that the Martini DPSM models show a reduced thickness compared to SM(18:0).

While the melting temperatures estimated with the biphasic approach, that is used during optimization for performance reasons, are not within the specified tolerance regime of 2 K but $\approx 5-6$ K below the target value and $\approx 3-4$ K below the lower target threshold, the new models are greatly improved compared to the current model, which was 20 K off target. Notably, the estimation of T_m is approach dependent. Estimations using the alternative, reversible melting approach with slow melting rates, based on Kowalik et al. [58] and Sun and Böckmann[59] (see SI for further details), which requires a very large computational budget (as done here, total simulation time for one T_m estimation > 90 μs) show an even better agreement with the experimental melting temperature.

The here-performed biphasic approach utilizes a bilayer that is half gel and half liquid. The gel phase is fabricated by quenching to a temperature well below the

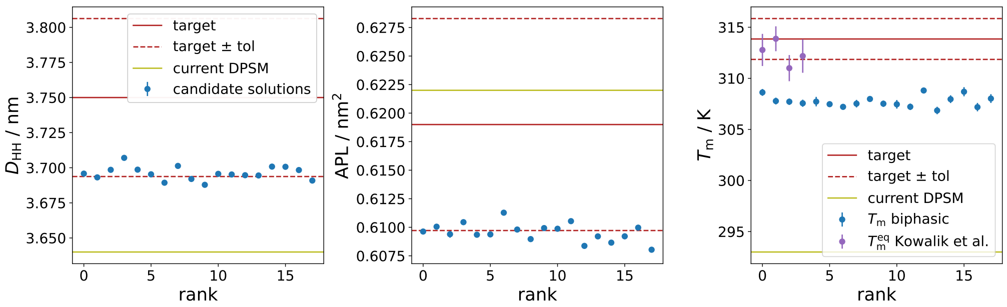


Figure 3.3: Thickness, average area per lipid and melting temperature for the set of statistically equal candidate solutions that remained after the second screen-to-the-best procedure performed after reevaluating the initial set 20 times.

melting temperature, and the gel phase system is combined with a preequilibrated liquid system. The combined system is then equilibrated with thermostats set to different temperatures for the two phases. As quenching and equilibration can take up to several hundreds of ns, reconstructing the starting structure for every candidate solution would significantly increase computational cost of a PSO run. Therefore, starting structures for this procedure were generated with the current DPSM parameters beforehand and equilibrated using the parameters of each candidate solution. While equilibration of the fluid phase is generally fast, this certainly is not the case for the gel phase. Considering that an unequilibrated phase is inherently less stable, the presence of an equilibrated liquid phase alongside an unequilibrated gel phase may lead to a slight systematic underestimation of the melting temperature (T_m) [60]. However, this potential underestimation can be anticipated and taken into account during the analysis.

The equilibrium melting rate approach does not suffer from this potential problem of unequally equilibrated phases. To minimize bias caused by the quenched starting structures used in this approach, for each validated candidate solution eight different starting conformations were generated.

3.4.1.2 Structural properties of the parameterized sphingomyelin models

Figure 3.4 shows the distributions of the newly parameterized bonds and angles for the candidate solutions in $\mathcal{P}^{\mathrm{g}}$. The atomistic target distributions are matched reasonably well in all cases. Some finer details of the atomistic model, like double peaks or extensive shoulders cannot be matched in the CG model. The parameterization philosophy of Martini 3 adopts a size-shape concept, where bond lengths are determined based on the molecular volume of the atomistic fragment mapped by the beads, rather than simply center of masses. This complication further underscores the necessity of employing multi-objective optimization algorithms to achieve effective molecule parameterization.

The solvent accessible surface area (SASA) is commonly used to further compare the molecular volumes and shapes between CG and AA models [11, 61]. Figure 3.5 shows the SASA values of $\mathcal{P}^{g}$ in comparison to the AA and current CG DPSM models. The SASAs are computed for the linker beads AM1 and AM2, as well as all supra-atoms that are directly connected to the linker, i.e., beads PO4, T1A, and C1B, as these connections are also parameterized. With SASA values of $\approx 6.24\,\mathrm{nm}^2$ all newly parameterized CG models show a better reproduction of the AA value ($5.24\,\mathrm{nm}^2$) compared to the current model ($6.45\,\mathrm{nm}^2$), but with discrepancy of $\approx 19\,\%$ all SASA values remain grossly too high. It appears that solely reparameterizing the linker region is not enough to fix this issue. Furthermore, using SASA directly as a target in the high-throughput optimization scheme is not necessarily beneficial, since a specific SASA value is not a unique representation of a certain shape. Therefore, comparisons of solvent accessible surface areas between AA and CG models are most helpful when done by simultaneous visual inspection. For automated parameterization, however, more detailed shape descriptors should be used.

3.4.1.3 Force field parameters

Non-bonded interactions: Due to the polar nature of the linker region of sphingolipids, only the chemical types of the P-block of the Martini 3 FF were eligible. As groups of 3 or 4 heavy atoms were combined into supra-atoms in the specified mapping, bead sizes small (S) and regular (default) could be chosen by the algorithm. Both bead sizes were permitted for both interaction sites, to allow for some wiggle room, even though 4 heavy atoms are grouped together into supra-atom AM1 and 3 into AM2. A slight miscount of mapped atoms is not uncommon in Martini, e.g., the mapping of the NC3 bead is actually 6-to-1 [10]. Generally, eligible bead types should be chosen with the Martini rules in mind. Martini's pragmatic philosophy allows for some freedom to match certain properties more accurately, but the bead type should not deviate strongly from the chemical identity of the molecule fragment[11].

One feature of the mixed-variable approach is that the optimization procedure directly yields a probability distribution of bead types, cf. Figure 3.6A. While for the interaction site AM2 there is clear consensus on the bead type, for AM1 only the size (small) is clearly determined, but there is some ambiguity regarding the interaction strength. The reduced size of one of the beads seems to be warranted, given the still too high SASA values shown above, and is also inline with the

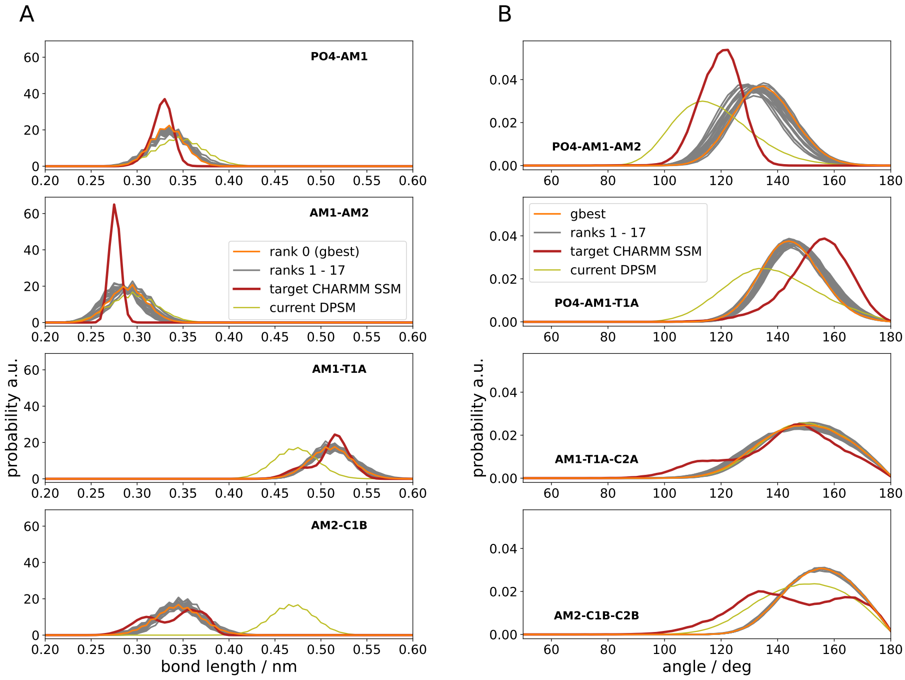


Figure 3.4: Validation of targets from rerun simulations for the set $\mathcal{P}^{\mathrm{g}}$. **A)** Bond length distributions. **B)** Angle distributions.

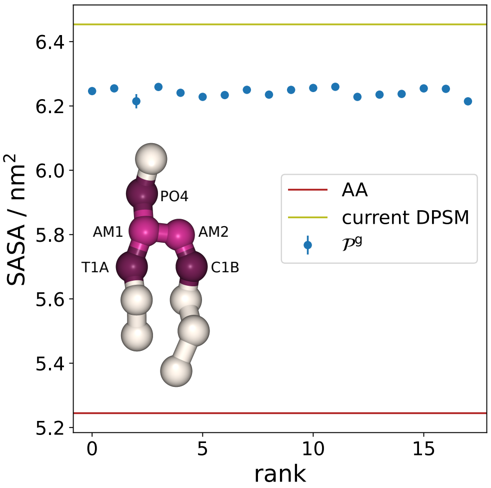


Figure 3.5: Solvent accessible surface area of the linker and beads connected directly to it. Beads involved in SASA calculation are highlighted.

new Martini 3 models of glycerolipids [11]. It is also worth mentioning that the chemical bead types chosen by our algorithm match the expected assignment suggested by Martini 3.

A converged "degenerate" probability distribution of bead types is the result of two or more bead types having indistinguishable effects on fitness. This can be caused by noise levels being larger than the fitness differences or the employed set of observables and training systems is lacking the necessary discriminatory power. Both issues can be remedied in post-optimization screening, but should optimally be addressed during optimization. As the former option would merely improve selection from the pool of generated candidate solutions, the later would potentially allow the generation of truly better solutions.

Additionally, for both, non-bonded and bonded FF parameters, diversity can be caused by the fact that the objective cost function for single valued observables (Eq. 3.7) has an error tolerance to accommodate for uncertainties in target data. With respect to these observables, different parameterizations with different "phenotypes" can have the same objective cost, as long as they are within the specified tolerances.

Bonded interactions: Table 3.4 lists the range of permitted bond parameters used in the optimization. The resulting bonded parameters of $\mathcal{P}^g$ are shown in Figure 3.6. For equilibrium bond lengths b_0 there is little variation between different candidate solutions. This strong consensus suggests that the optimization has converged and that small changes in equilibrium bond length are linked to significant cost changes. The situation for the force constants is quite different. The values fluctuate over a relatively large range, compared to the predefined domain of permitted values. The measured bond length distributions (Figure 3.4A) show that these seemingly substantial differences in force constant values have only minor effects on the molecule's behavior.

The situation for the angle FF parameters is similar. The equilibrium values show smaller variances than the force constants, compared to their respective domain sizes of applicable values. Again, the differences in FF parameters have little effect on the observed distributions (cf. Figure 3.4B). Notably, the optimal force constants for the angles PO4-AM1-T1A and AM2-C1B-C2B were close to or at the maximum of their permitted ranges. Further optimization was therefore likely hindered, and a wider range should have been chosen.

In a similar vein to the discussion surrounding non-bonded parameters, the relatively wide range of force constants in $\mathcal{P}^g$ indicates that additional metrics or training systems could be employed to further optimize the overall performance of candidate solutions while maintaining the quality of the employed observables. For instance, exploring lipids in environments other than a bilayer, which induce

bond	GROMACS bond func. type	b_0 / nm	fc / kJ/mol/nm^2
PO4-AM1	1	0.25 – 0.40	1000 – 9000
AM1-AM2	1	0.20 – 0.35	1000 – 9000
AM1-T1A	1	0.40 – 0.55	1000 – 9000
AM2-C1B	1	0.25 – 0.50	1000 – 9000
angle	GROMACS angle func. type	a_0 / deg	fc / kJ/mol
PO4-AM1-AM2	2	90 – 180	5 – 100
PO4-AM1-T1A	2	90 – 180	5 – 100
AM1-T1A-C2A	2	180	5 – 100
AM2-C1B-C2B	2	180	5 – 100

Table 3.4: Bonded interactions. GROMACS function type; permitted parameter ranges for equilibrium bond length / angle, and corresponding force constants.

different lipid conformations, could benefit from a candidate solution with a lower angle force constant to allow for increased conformational variation.

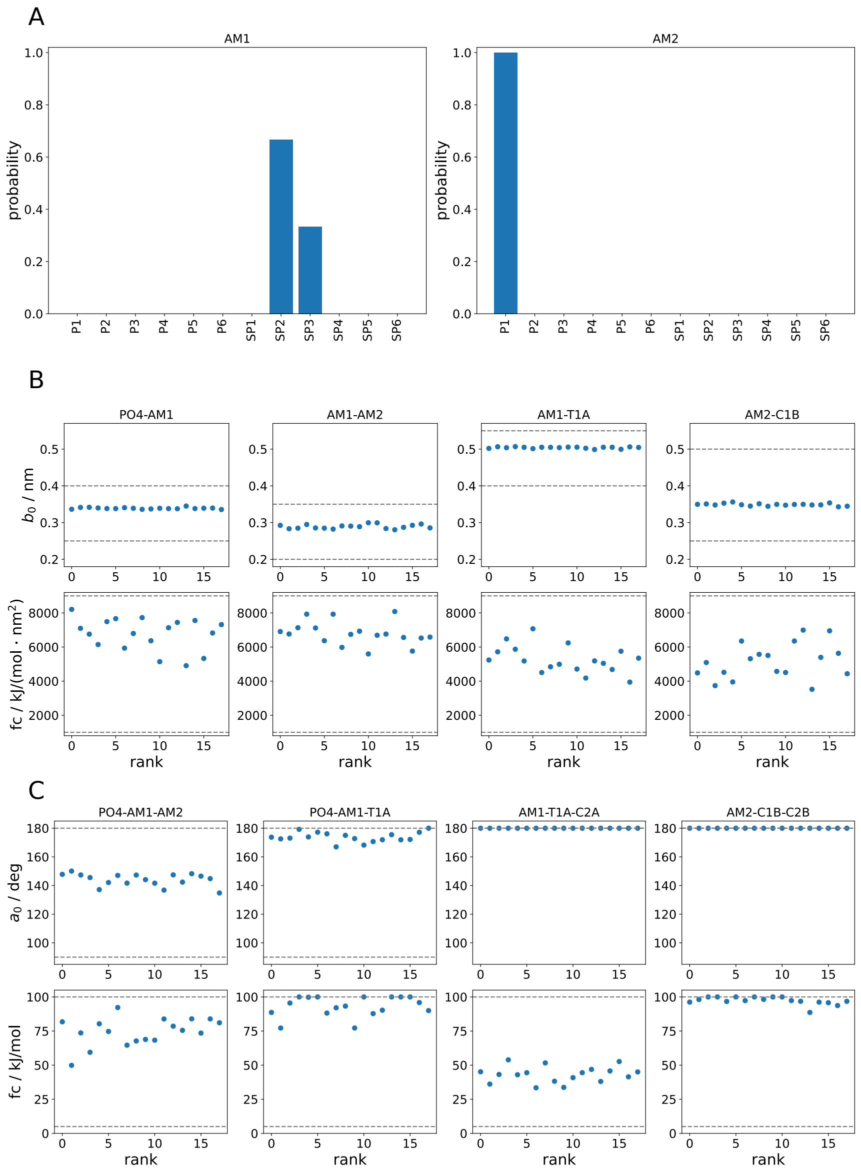


Figure 3.6: Force field parameters of the set of statistically equivalent solutions $\mathcal{P}^g$ for the sphingolipid linker region. **A)** Bead probability distributions. **B)** Bond parameters. Dashed lines are upper and lower parameter limits. **C)** Angle parameters. Dashed lines are upper and lower parameter limits. The equilibrium angles of AM1-T1A-C2A and AM2-C1B-C2B are not varied during optimization. They are fixed at 180°.

3.4.2 Noise-mitigation improves quality of parameterized models

We investigated whether the simple noise-mitigation strategy described in Section 3.2.2.1 can improve the quality of solutions found by the algorithm. The swarm size, training systems, observables and weights are the same as in Section 3.4.1. We tested three different resampling allocation settings and compared these to the mv-PSO without noise-mitigation. Each optimization run was given a fixed computational budget of 16128 MD simulation slots.

With the given number of MD simulation slots, a swarm size of 64 particles, and 3 training systems required for one full objective function evaluation, this amounts to 84 iterations for the mv-PSO without resampling (named mv-PSO-R0). In the optimization runs with resampling an initial computational budget of $64 \cdot 3 = 192$ MD simulation slots was used for one full objective function evaluation of each particle, and a second equally sized computational budget was allocated to reevaluate the melting temperature (the target observable with the largest variance) of the best 16, best 32, or all 64 candidate solutions of the current iteration. For brevity we will refer to these as mv-PSO-R16, mv-PSO-R32, and mv-PSO-R64. Due to the fixed computational budget, for each particle involved in resampling, T_m was reevaluated 12, 6, or 3 times. As half of the total computational budget was used for resampling, the number of iterations was set to 42 in these runs.

From the literature on PSO noise-mitigation[31, 62] we draw the expectation that which of the resampling, or no resampling, strategies is the best, depends on the level of noise. If noise levels are very low, the additional number of possible iterations, when forgoing resampling, could lead to better solutions. For intermediate noise levels, initial fitness evaluation results in a sufficient differentiation of good and bad solutions, i.e., overall sorting is roughly correct, and the focus on improving sorting of the very best solutions is most helpful. In case of even higher noise levels initial sorting would be vastly incorrect and a larger fraction of the swarm needs to be resampled to achieve satisfactory overall sorting. As a consequence, the sorting quality of the very top would be degraded, as there is less computational budget allocated here.

The true quality of a candidate solution is not necessarily reflected by the cost estimated during an optimization run, as there is some uncertainty in estimates of target observables other than T_m, and the confidence level of the T_m estimation with different resampling settings differs vastly. Therefore, validation is required. As we are mostly interested in the quality verification of the best solutions, the first step of the screen-to-the-best procedure from Boesel et al. [34] can be used to select the statistically equivalent set of candidate solutions. For mv-PSO-R16 the set $\mathcal{P}_t^g$ contains 69 candidate solutions. Due to the increased uncertainty in mv-PSO-R32 and mv-PSO-R64, their respective sets $\mathcal{P}_t^g$ contain hundreds of candidate solutions. To keep the computational cost for validation manageable, we selected only the 72 best solutions of these optimization runs for validation. As there are no variance estimates in the optimization run without resampling, the selection procedure is not applicable. Again, the 72 best solutions from the optimization run were selected for validation. All candidate solutions chosen for validation were fully (all training systems, all observables) reevaluated 20 times. The resulting rerun cost vs. the originally estimated cost is shown in Figure 3.7. Clearly, mv-PSO-R16 gave the best results, while the quality of the best solutions in the three other cases does not differ much. Furthermore, the fact that for all selected candidate solutions of mv-PSO-R0 the rerun cost estimate is substantially higher than the original cost estimate indicates that these original estimate are particularly favorable. While there are also candidate solutions with substantial differences in original and rerun cost for the resampling systems – in this case mostly caused by APL fluctuations – these are much less frequent and there is much better correlation between original and rerun cost (Pearson correlation coefficient 0.21 vs. 0.64, for mv-PSO-R0 and mv-PSO-R16, respectively).

Our interpretation of these results is the following: The noise level is low enough, so that even without noise-mitigation, the sorting of candidate solutions is correct in a coarser sense and the swarm is guided towards the "correct" vicinity in parameter space. Yet, noise levels are substantial enough, so that resolution of finer cost differences is impeded. Only the concentrated allocation of the resampling budget on the top 16 solutions lowers the cost estimation errors sufficiently such that improved candidate solutions can be found.

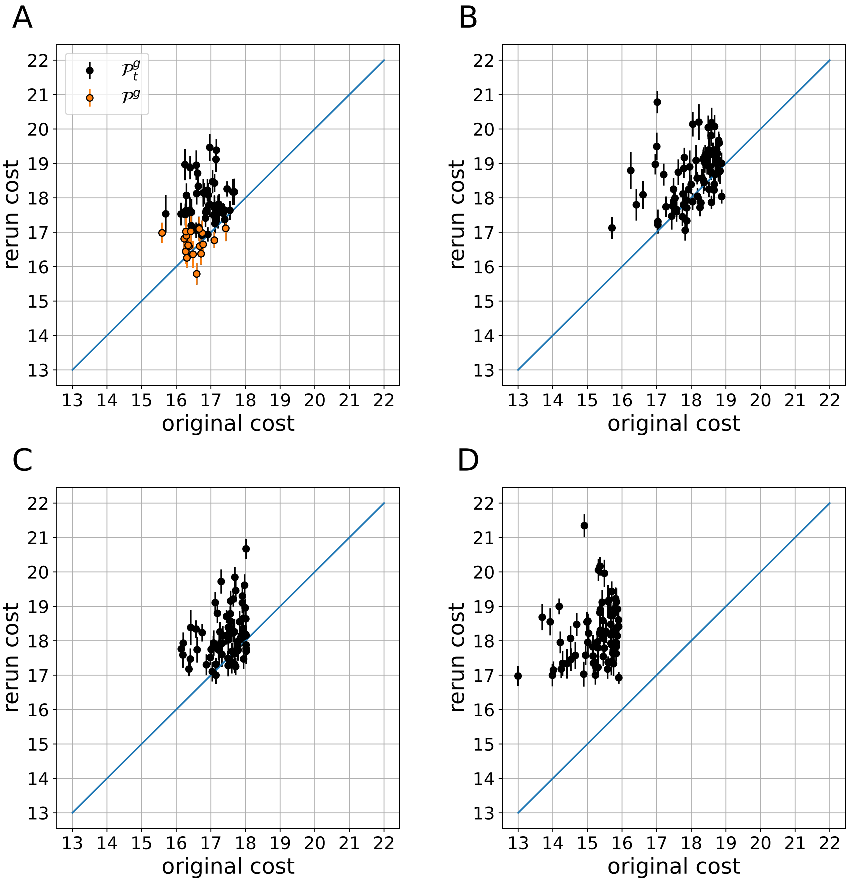


Figure 3.7: Comparison of cost estimated during the optimization run and average cost estimated from repeated reruns of $\mathcal{P}_t^{\mathrm{g}}$ in **(A)** and the 72 best candidate solutions in **(B)**-**(D)**. Error bars are standard errors. **(A)** Original cost: 16 particles resampled, 1+12 T_{m} samples. **(B)** Original cost: 32 particles resampled, 1+6 T_{m} samples. **(C)** Original cost: 64 particles resampled, 1+3 T_{m} samples. **(D)** No resampling during optimization, but twice as many iterations

3.4.3 Validation: Phase behavior of binary sphingomyelin-cholesterol membranes

To discern the universality of the parameterization, we conducted a validation test on a target that was not included in the optimization process. Specifically, we assessed whether the optimized model (ranked 0 within the set $\mathcal{P}_\mathrm{g}$) could accurately replicate the phase behavior of binary sphingomyelin-cholesterol membranes. Experimental results show that below T_m, increasing cholesterol content fluidizes the otherwise frozen systems[63]. For very low cholesterol concentrations the system remains in the gel phase ($\mathrm{S_o}$), at around 10 mol% there is a transition to coexistence of gel and liquid ordered ($\mathrm{L_o}$) domains, and above $\approx$ 30 mol% there is a $(\mathrm{S_o}+\mathrm{L_o})/\mathrm{L_o}$ transition [63]. As can be seen in Figure 3.8 the optimized model correctly reproduces the experimental findings, while systems simulated with the current DPSM model are always in the fluid phase, regardless of the cholesterol concentration. These findings therefore highlight the robust universality and transferability of the parameterization acquired with GCCompiler.

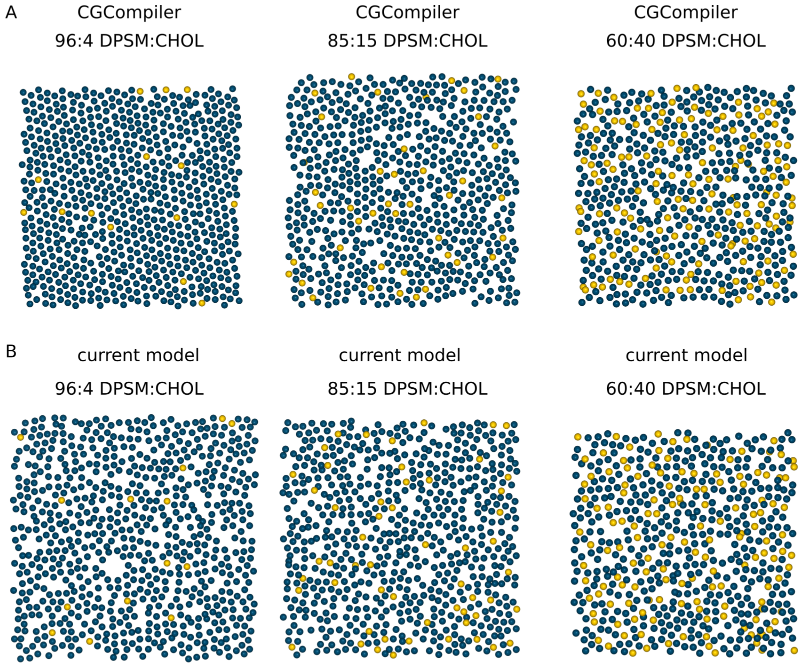


Figure 3.8: Phase behavior of binary sphingomyelin-cholesterol membranes. T = 300 K. Production simulations were 1 μs. Snapshots are from the last frame. To help with the nucleation of the gel phase, all systems were pre-equilibrated for 50ns at 290K (CGCompiler result) or 270K (original DPSM). A) CGCompiler optimized (rank 0 of $\mathcal{P}_\mathrm{g}$) B) current DPSM

3.5 Discussion & Conclusion

We have illustrated how to apply mixed-variable particle swarm optimization for automated CG molecule parameterization. As an example application, we parameterized the sphingolipid linker region for the Martini 3 FF. The newly parameterized sphingomyelin model reproduces important target observables accurately, including the melting temperature, which was $\approx 20\,\mathrm{K}$ off target before and is now within $\approx 2\,\mathrm{K}$ of the experimental reference. Notably, reproduction of experimental melting temperatures had been historically problematic in Martini lipid models[64].

The mixed-variable approach offers a major advantage when parameterizing molecules for building-block force fields. Due the explicit use of building blocks, every candidate model is a valid parameterization in the given FF. Otherwise, changing non-bonded interaction parameters of the FF's building blocks breaks the validity of their parameterization. Candidate solutions generated by a continuous treatment of non-bonded interactions have to be converted to a valid FF model, followed by additional validation of this model.

A drawback of the mixed-variable treatment is that some advanced improvements to PSO, such as fuzzy parameter tuning of Nobile et al.[65], are not directly applicable to mv-PSO, because in the categorical representation there is no similarity metric, which is utilized in the PSO parameter tuning. This could be overcome by using discrete ordered representation for non-bonded interactions instead of the categorical treatment.

One of the great benefits of automated parameterization algorithms is the simultaneous optimization against multiple structural and thermodynamic target data. As thermodynamic observables can be expensive to estimate accurately in MD simulations, the formal consideration of noise in objective function values is an important conceptual improvement. As demonstrated, optimization with applied noise-mitigation produced significantly better solutions and the utilized screen-to-the-best procedure provides a systematic approach to the post-optimization selection of the best model.

Although we have demonstrated the adverse effects of objective function value noise on the sorting and performance of PSO, it is important to note that the non-deterministic nature of particle swarm optimization necessitates multiple repetitions of full optimization runs to confidently determine the most effective noise-mitigation setting. Achieving a high level of confidence in identifying the optimal approach would require a significant number of iterations. Furthermore, the 'ground truth', i.e., the true score of a candidate parameterization, is unknown, hence a large amount of validation simulations would be required. This is not feasible, due to a high computational cost. Rigorous development and testing of noise-mitigation strategies should not be done with objective function evaluations that require costly MD simulations, and are therefore beyond the scope of this paper. Moreover, the additionally gained insight, would only be of moderate value. The PSO literature has shown that under significant noise PSO performance is degraded and performance differences between resampling methods for noise-mitigation are problem and noise-level dependent. Generally, noise-mitigation methods employing OCBA perform the best under various circumstances[31, 66], but its sequential secondary budget allocation puts constraints on the parallelization of the parameterization algorithm. Still, its integration into the parameterization pipeline should be explored in the future.

Together with the general benefits of automation, the here-presented conceptual advantages will further facilitate rigorous CG molecule parameterization. The CGCompiler Python package that comes with our method is tailor-made for parameterization tasks in building-block FFs, such as Martini. Also larger building blocks, i.e., a molecule class with shared regions can be parameterized simultaneously. Our approach is not limited to lipid parameterization, but can be applied to any kind of molecule. CGCompiler can be easily adapted to the needs of a specific parameterization task. Implementing new observables is not much different from writing Python functions for analyzing MD data. Importantly, our automation platform eases collaborations between individual researchers since a clear overview of the parameterization flow is provided. This also renders force-field reproducibility as well as retrospective force-field corrections, such as corrections to the targets (e.g., improved atomistic force-fields or simulation settings) or inclusion of additional targets rather straightforward.

The here-presented study focuses on method development and the sphingolipid linker parameterization was merely a test case. The parameters of the head group and lipid tails, predefined in our study, are still actively improved/(re)parameterized by the core developers [11]. Once these final parameters are released, reparameterization of the linker may be necessary, ideally with an even broader set of training systems, including liquid ordered-disordered phase behavior.

Properly defining the set of feasible bead type choices, for the fragments that are to be optimized with CGCompiler, is a crucial step in the parameterization of a molecule. In the Martini FF, bead type assignment is based on partitioning data of isolated beads [10], and as of Martini 3 also partitioning of whole molecules and miscibility data are considered [11]. The Martini 3 supporting material lists defaults bead type choices [11]. For more accurate bead type assignment proximity and connectivity effects between fragments need to be considered, and perturbations around the default solution are therefore allowed [11]. In complex cases, bead type selection can become non-trivial when several proximity effects are present in a molecule [11]. The use of target data other than the free energies of transfer is recommended and regularly employed [54, 61, 67–71] when refining bead type choices. In our proof of concept parameterization, we have chosen to use the full range of P-block beads to showcase the capabilities of the algorithm. As the free energy of transfer was not explicitly part of the loss function, this choice could have possibly resulted in a deviation of free energies of transfer in the order of a few kJ/mol per linker fragment. As the final best bead type choices closely match the default bead type choices, this is not an issue for the optimized CG model of sphingomyelin. In a normal parameterization run and when a fragment's partitioning and miscibility behavior is encoded by the choice of possible bead types and not explicitly part of the loss function, it is recommended to restrict the set of feasible bead types more narrowly. Otherwise, if applicable to the molecule that is to be parameterized, researchers should consider including the free energy of transfer into the loss function, either by calculating the free energy of transfer for the whole molecule, or by making use of partitioning data for individual fragments.

In order to achieve fully automated molecule parameterization in high-throughput applications, the development of an automated mapping and selection of bonded terms remains a crucial component. Currently, mapping and parameter optimization are separate tasks, but integrating an automated mapping scheme into the parameterization pipeline could be facilitated prior to employing mixed-variable particle swarm optimization, utilizing CGCompiler. The choice of bonded parameters not only influences the accuracy of the model but also impacts simulation stability. Various strategies, such as the use of virtual sites, restricted bending potentials, hinge and "divide and conquer" constructions [68, 72], have been previously described to address instability. Additionally, careful consideration of constraints is necessary to ensure simulation stability and prevent artificial temperature gradients [73, 74]. These aspects should be incorporated as essential steps in a future fully automated parameterization pipeline.

Reweighting of CG trajectories could be an interesting route to decrease the computational effort required for parameterization [75–77], particularly in a high-throughput setting. However, this currently is not part of CGCompiler for the following reason. The applicability of reweighting critically depends on the overlap of the original and the reweighted trajectory [75, 76]. As the candidate solutions in the swarm at a given iteration can have rather different potentials, it is unknown beforehand for how many candidate solutions reweighting can be applied and for how many a new CG trajectory has to be generated. As CGCompiler is intended to be used with a high degree of parallelization on compute clusters, where a compute job runs on a fixed hardware allocation, not having to run a simulation for some of the candidate solutions does not directly result in decreased usage of a computational budget. For reweighting to be of use, an adaptive scheduling algorithm would be required, which could be implemented in future versions of CGCompiler.

When linearly increasing the number of optimized parameters, the search space grows exponentially, which negatively affects convergence of the optimization algorithm. In the here presented study 2 categorical (non-bonded) and 14 continuous (bonded) parameters were optimized simultaneously. In a recent reparameterization of PC lipid tails [16] 77 bonded parameters were calibrated using a different flavor of PSO. As both parameterizations required only moderate swarm sizes and number of iterations for convergence, we expect that our PSO approach can be used for the parameterization of larger molecules as well. However, for very large molecules with several hundreds or even thousands of unique parameters, parameterization with CGCompiler or similar approaches likely becomes unfeasible. On one hand standard PSO in general is not the method of choice to tackle such Large Scale Optimization Problems (LSOP) [78]. On the other hand, even if the PSO part of CGCompiler would be replaced by an optimization algorithm more suitable for an LSOP, the number of required function evaluations, i.e., MD simulations, likely remains too large to be of practical use in a molecule parameterization task.

No matter the number of parameters that are co-optimized, in order to lessen the computational cost, convergence can be facilitated by restricting the search space. Optimization with CGCompiler must then be performed on an initial, close guess, rather than scanning a broad parameter range. Such an initial guess can be constructed either manually by following the Martini 3 rule book, or by an automated tool. (Auto-Martini [18] and the method of Potter et al. [19] would need to be adapted for Martini 3, in order to be used in such a parameterization pipeline.) A not too narrow restriction of search space will not hinder discovery of good solutions, as parameters that are very far away from the standard Martini rules are not of interest anyway. Bond lengths that are very different to the atomistic reference would result in misshaped molecules. Very different bead types would, for example, result in incorrect partitioning behavior. In principle, these unwanted regions of the search space are filtered out by the cost function, but they can be excluded beforehand to save computational effort. A narrower search space restriction is expected to be more important when the number of parameters is large.

Another future prospect is the advancement of true non-scalarized multi-objective optimization, which eliminates the need for user-defined weights on the targets within the objective function. However, it can also be argued that these user-defined weights, which reflect the importance of targets based on intuition, experience, or additional knowledge, along with the predefined set of relevant structural and thermodynamic targets for the CG force-field, encompass what is commonly known as the "force-field's philosophy". In this sense, the user-defined weights embody the guiding principles that shape the force-field.

Appendix 3.A Supporting Information

3.A.1 Melting temperature

Due to the slow kinetics of the gel-liquid phase transition, estimating the melting temperature can be difficult to estimate in simulations [60]. For use in automated parameterization methods, where the melting temperature of many candidate solutions has to be estimated, the trade-off between accuracy and computational cost is of particular high importance. Preparing the system in stripes, i.e., half gel and half fluid (cf. Figure 3.9A), bypasses the slowest step in the transition, the nucleation[60]. By simulating the biphasic system at a range of temperatures, observing the direction and rate of domain growth, and fitting rates to an Arrhenius-like equation, Coppock and Kindt have estimated T_m for atomistic DPPC and DPSM [60]. In tests performed by us with Martini DPPC and DPSM, this procedure did not provide results reliable enough for application in a high-throughput manner. A similar biphasic system setup was used by Carpenter et al. [79] in Martini lipid refinement. Instead of fitting domain growth rates, they used the area per lipid as a proxy for which phase prevails at a certain temperature, utilizing the fact that the highly ordered tails in the gel phase result in a much smaller area per lipid compared to the liquid phase. Phase identification with this procedure is very fast and reliable at temperatures more than a few Kelvin away from the transition temperature. The melting temperature is then given as a range between the highest temperature where the system converges to a gel phase and the lowest temperature the system ends up in the liquid phase. The accuracy of this method is strongly influenced by the employed temperature-spacing. Close to the transition temperature, longer and repeated simulations are necessary, due to the stochasticity of the melting/freezing process (cf Figure 3.9B).

The area per lipid has a positive linear relationship with temperature and a sharp increase at the melting temperature, as shown in Figure 3.10a. Therefore, to estimate the melting temperature from the temperature-dependent APL data, we fit the sigmoidal function

$$\mathrm{APL}(T) = \mathrm{APL}_0 + c \cdot T + \frac{\Delta\mathrm{APL}}{(1 + \exp(-k \cdot (T - T_\mathrm{m})))} \tag{3.11}$$

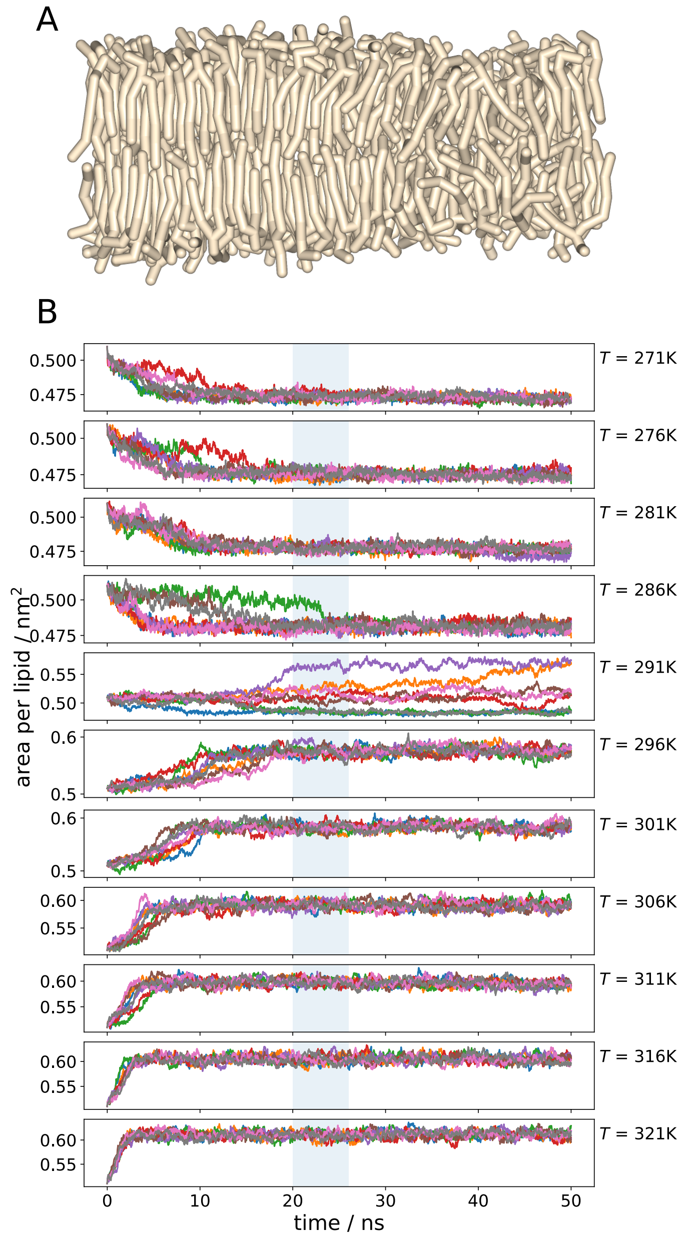
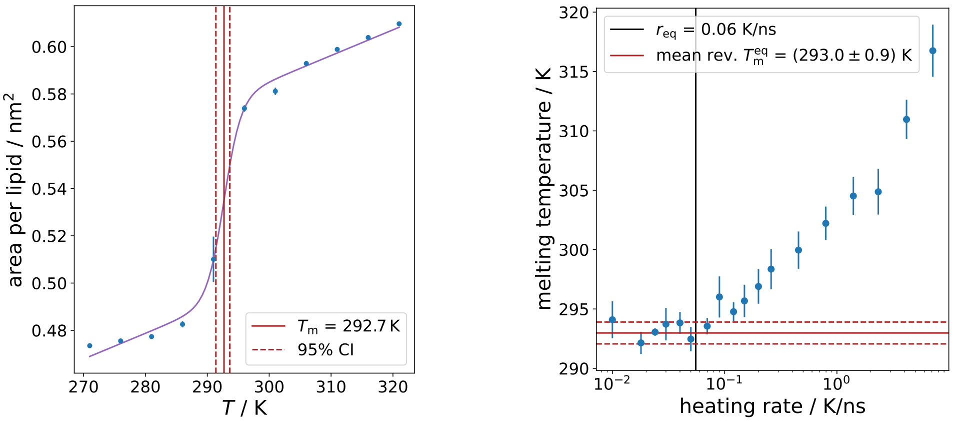


Figure 3.9: Melting temperature estimation. **A**: Snapshot of initial configuration of a DPSM bilayer, where half of the lipids is in the gel phase, while the other half is in the fluid phase. **B**: Area per lipid vs. time at different temperatures. The shaded area represents the time window over which the area per lipid is averaged for use in the fit of Eq. 3.11. Production simulations during PSO are typically 25 ns long.



(a) Tm APL fit, T_m = 292.7 K with a 95% CI of [291.4 K, 293.6 K]

(b) melting temperature. [58]

Figure 3.10: Estimation of melting temperatures with two independent methods for the old DPSM model.

where T is the temperature, APL$_0$ is the theoretical area per lipid at $T = 0\,\mathrm{K}$, c is the slope in the linear regime, ΔAPL describes the height of the APL jump at the melting temperature T_m, and k determines the broadness of the transition.

When using this method in an automated parameterization setting, the case that T_m might be outside of the predefined temperature range has to be handled properly. To this end we additionally fit a line to the APL(T) data and use the Akaike information criterion (AIC)[80, 81] with the modification for small sample sizes (AIC$_\mathrm{c}$)[82] to determine which model (sigmoidal or linear) is a better description of the data. If the linear model is better, i.e., it has a lower AIC$_\mathrm{c}$, and all APL values are above or below a threshold (the average initial APLs), T_m is considered to be out of range and T_m is set to a particular low or high value, respectively. Hereby, candidate solutions with a melting temperature far off the target value receive a high cost value in the PSO. If the sigmoidal model is selected, or the linear model is a better fit but the APL values cross the threshold, T_m is taken from the sigmoidal fit. A few typical examples of this procedure are shown in Figure 3.11.

All of the above biphasic methods are sensitive to the construction of the stripe structure. In particular, an improperly equilibrated gel phase can lead to an underestimation of T_m [60]. An alternative, independent approach to estimate T_m is based on a two-state kinetic rate model from Kowalik et al [58]. In this approach a system in the gel phase is heated with different heating rates. According to the two-state kinetic rate model, the melting process can be divided into regimes of

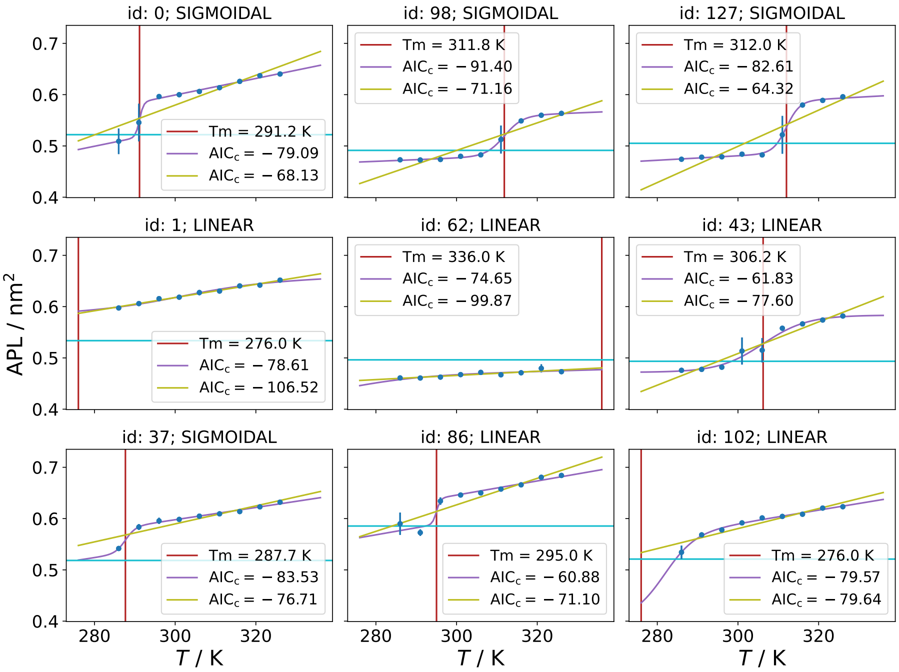


Figure 3.11: Examples of fit model selection in the biphasic approach to estimate T_m.

reversible and irreversible melting. In the reversible melting regime, for slow heating rates, the system is assumed to be close to thermal equilibrium and melting and freezing can both occur. In this regime the apparent melting temperature is independent of the heating rate r, i.e., $T_\mathrm{m}^\mathrm{app}(r) \approx T_\mathrm{m}^\mathrm{eq}$. For fast heating rates, melting is assumed to be irreversible. In the irreversible melting regime, the two-state model predicts a dependency of the apparent melting temperature on the heating rate which can be approximated by $T_\mathrm{m}^\mathrm{app}(r) \propto \ln r$. Both regimes are divided by a characteristic heating rate r_eq.

Kowalik et al. used a series of melting simulations with fast heating rates, i.e., in the irreversible regime, to obtain several $T_\mathrm{m}^\mathrm{app}(r)$ values, determine the characteristic melting rate r_eq, and finally extrapolate the equilibrium melting temperature T_m^eq. Based on the two-state kinetic rate model from Kowalik et al., Sun and Böckmann [59] simply used a broad range of heating rates, including the reversible regime. The equilibrium melting temperature was calculated by averaging over the $T_\mathrm{m}(r)$ in the reversible regime, i.e., $r < r_\mathrm{eq}$.

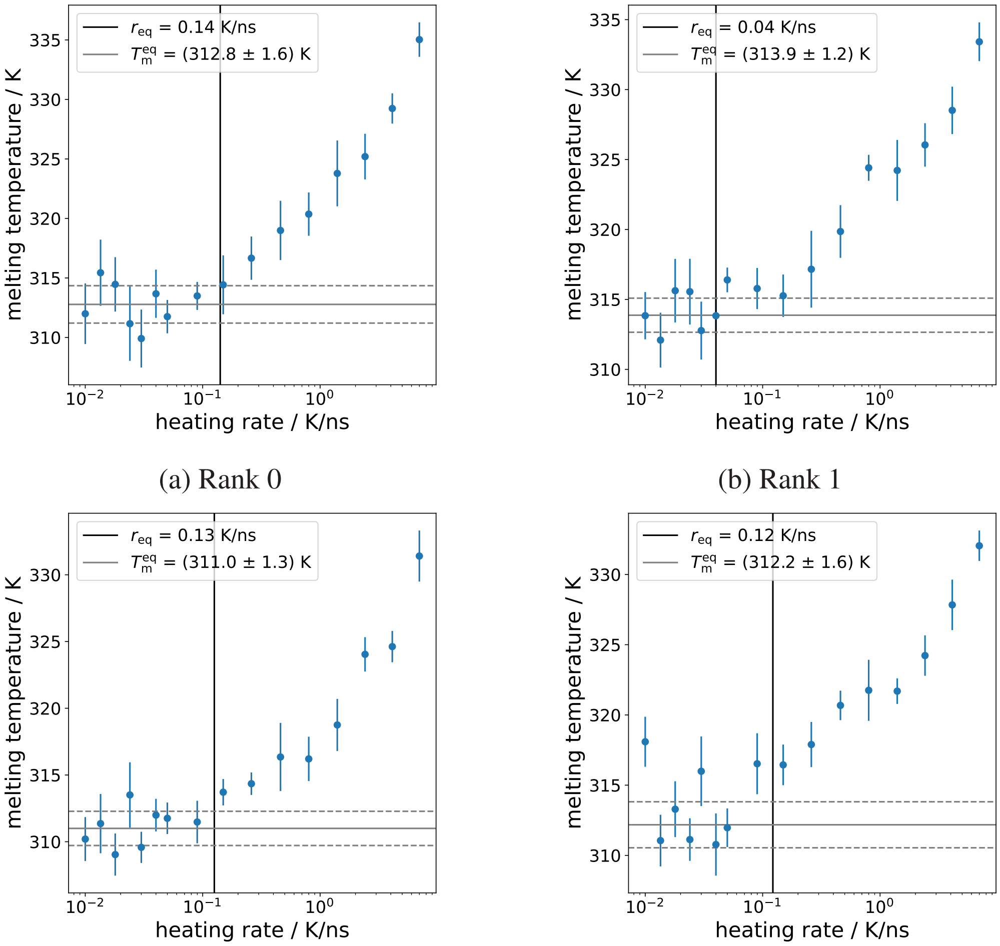
Due to the slow rates and concomitant long simulation times, we use this approach only for validation. To minimize bias caused by the quenched starting conformations we typically generate eight independent conformations for each validated candidate solution.

The rate dependent melting temperatures in this approach are obtained by fitting

$$H(T) = H_0 + c_p \cdot T + \frac{\Delta H}{(1 + \exp(-k \cdot (T - T_\mathrm{m})))} \tag{3.12}$$

where H is the enthalpy, c_p the heat capacity, H_0 is the enthalpy at $T = 0\,\mathrm{K}$, ΔAPL describes the height of the APL jump at the melting temperature T_m, and k determines the broadness of the transition. The functional form is the same as in Eq. 3.11. Figure 3.10b shows results of this approach for the old DPSM model. The heating rate dependency matches the prediction of the two-state model. Comparision of Figures 3.10a and 3.10b show that the melting temperatures for the old DPSM model, obtained with both approaches, are in good agreement.

3.A.1.1 Melting temperature validation of 4 best candidate solutions



(a) Rank 0

(b) Rank 1

(c) Rank 2

(d) Rank 3

Figure 3.12: T_{m} of the 4 best candidate solutions with the reversible melting approach.

3.A.2 Noise

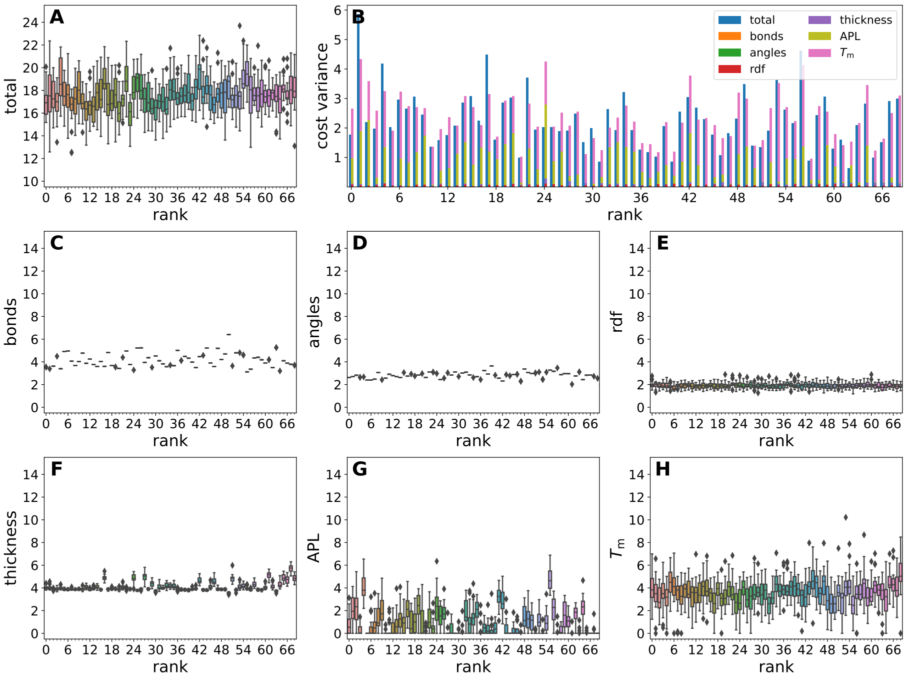
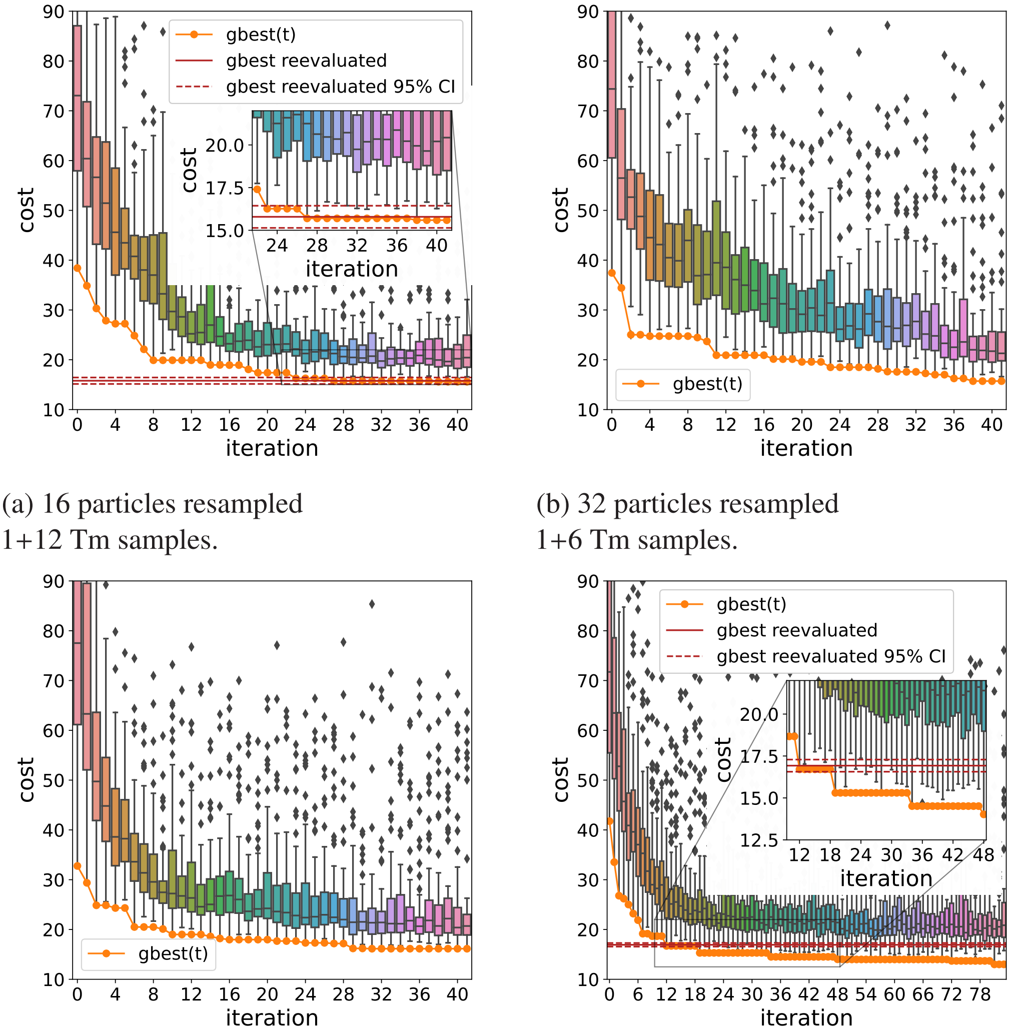


Figure 3.13: Noise levels of individual observables.



(a) 16 particles resampled 1+12 Tm samples.

(b) 32 particles resampled 1+6 Tm samples.

(c) 64 particles resampled 1+3 Tm samples.

(d) No resampling during optimization. But twice as many iterations

Figure 3.14: Comparison of cost evolution during optimization with and without noise-mitigation through resampling.

3.A.3 DPSM-CHOL 2d center-of-mass radial distribution function

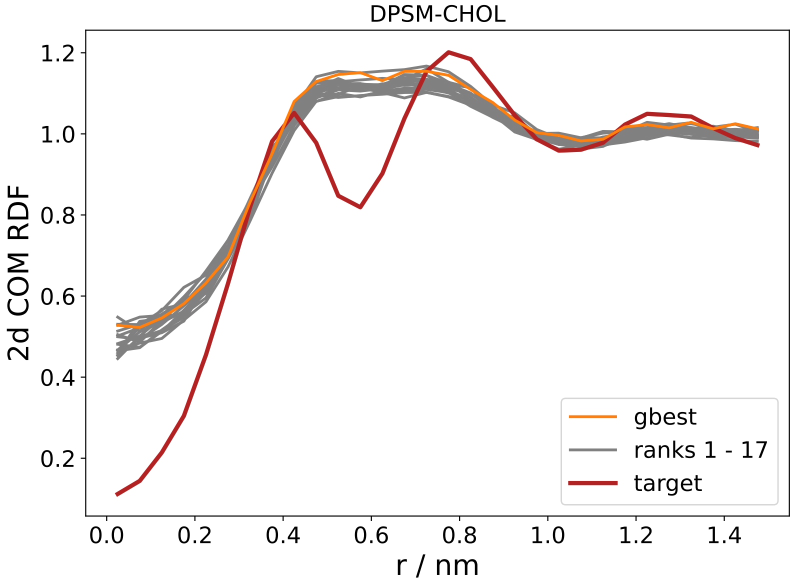


Figure 3.15: 2d COM radial distribution function (RDF). The distance *r* is measured in the x-y plane, i.e., parallel to the membrane. The 2d RDF is calculated per leaflet and averaged.

3.A.4 DPSM topology for Gromacs

```
[ moleculetype ]
; molname    nrexcl
DPSM      1

[ atoms ]
; id     type   resnr  resname atomname cgnr charge
   1       Q1       1  DPSM    NC3      1   1.0
   2       Q5       1  DPSM    PO4      2  -1.0
   3      SP2       1  DPSM    AM1      3   0.0
   4       P1       1  DPSM    AM2      4   0.0
   5       C1       1  DPSM    T1A      5   0.0
   6       C1       1  DPSM    C2A      6   0.0
   7       C1       1  DPSM    C3A      7   0.0
   8       C1       1  DPSM    C1B      8   0.0
   9       C1       1  DPSM    C2B      9   0.0
  10       C1       1  DPSM    C3B     10   0.0
  11       C1       1  DPSM    C4B     11   0.0

[ bonds ]
;   i    j    funct   r0   fc
    1     2   1   0.40000   7000
    2     3   1   0.33632   8207
    3     4   1   0.29241   6909
    3     5   1   0.50190   5239
    5     6   1   0.47000   3800
    6     7   1   0.47000   3800
    4     8   1   0.34964   4483
    8     9   1   0.47000   3800
    9    10   1   0.47000   3800
   10    11   1   0.47000   3800

[ angles ]
; i   j   k   funct   theta0  fc
    2     3     4   2   147.805  81.78
    2     3     5   2   173.718  88.64
    3     5     6   2   180.000  45.16
    5     6     7   2   180.000     35
    4     8     9   2   180.000  96.23
    8     9    10   2   180.000     35
    9    10    11   2   180.000     35
```

References

[1] Risselada, H. J. and Marrink, S. J. “The molecular face of lipid rafts in model membranes”. In: *Proceedings of the National Academy of Sciences* vol. 105, no. 45 (2008), pp. 17367–17372.

[2] Marrink, S. J., Risselada, J., and Mark, A. E. “Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model”. In: *Chem .Phys. Lipids* vol. 135, no. 2 (2005), pp. 223–244.

[3] Risselada, H. J. and Marrink, S. J. “The freezing process of small lipid vesicles at molecular resolution”. In: *Soft Matter* vol. 5, no. 22 (2009), pp. 4531–4541.

[4] Lyubartsev, A. P. and Laaksonen, A. “Calculation of effective interaction potentials from radial distribution functions: A reverse Monte Carlo approach”. In: *Phys Rev E* vol. 52, no. 4 (1995), p. 3730.

[5] Lyubartsev, A. P. “Multiscale modeling of lipids and lipid bilayers”. In: *Eur. Biophys. J.* vol. 35, no. 1 (2005), pp. 53–61.

[6] Izvekov, S. et al. “Effective force fields for condensed phase systems from ab initio molecular dynamics simulation: A new method for force-matching”. In: *J. Chem. Phys.* vol. 120, no. 23 (2004), pp. 10896–10913.

[7] Izvekov, S. and Voth, G. A. “A multiscale coarse-graining method for biomolecular systems”. In: *J. Phys. Chem. B* vol. 109, no. 7 (2005), pp. 2469–2473.

[8] Lafitte, T. et al. “Accurate statistical associating fluid theory for chain molecules formed from Mie segments”. In: *J. Chem. Phys.* vol. 139, no. 15 (2013), p. 154504.

[9] Papaioannou, V. et al. “Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from Mie segments”. In: *J. Chem. Phys.* vol. 140, no. 5 (2014), p. 054107.

[10] Marrink, S. J. et al. “The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations”. In: *The Journal of Physical Chemistry B* vol. 111, no. 27 (2007), pp. 7812–7824.

[11] Souza, P. C. T. et al. “Martini 3: a general purpose force field for coarse-grained molecular dynamics”. In: *Nat. Methods* vol. 18, no. 4 (2021), pp. 382–388.

[12] Alessandri, R. et al. “Pitfalls of the Martini Model”. In: *J. Chem. Theory Comput.* vol. 15, no. 10 (2019), pp. 5448–5460.

[13] Risselada, H. J. “Martini 3: a coarse-grained force field with an eye for atomic detail”. In: *Nat. Methods* vol. 18, no. 4 (2021), pp. 342–343.

[14] Graham, J. A., Essex, J. W., and Khalid, S. “PyCGTOOL: Automated Generation of Coarse-Grained Molecular Dynamics Models from Atomistic Trajectories”. In: *J. Chem. Inf. Model.* vol. 57, no. 4 (2017), pp. 650–656.

[15] Empereur-Mot, C. et al. “Swarm-CG: Automatic Parametrization of Bonded Terms in MARTINI-Based Coarse-Grained Models of Simple to Complex Molecules via Fuzzy Self-Tuning Particle Swarm Optimization”. In: *ACS Omega* vol. 5, no. 50 (2020), pp. 32823–32843.

[16] Empereur-mot, C. et al. “Automatic Optimization of Lipid Models in the Martini Force Field Using SwarmCG”. In: *J. Chem. Inf. Model.* vol. 63, no. 12 (2023), pp. 3827–3838.

[17] Empereur-mot, C. et al. “Automatic multi-objective optimization of coarse-grained lipid force fields using SwarmCG”. In: *J. Chem. Phys.* vol. 156, no. 2 (2022), p. 024801.

[18] Bereau, T. and Kremer, K. “Automated Parametrization of the Coarse-Grained Martini Force Field for Small Organic Molecules”. In: *J. Chem. Theory Comput.* vol. 11, no. 6 (2015), pp. 2783–2791.

[19] Potter, T. D., Barrett, E. L., and Miller, M. A. “Automated Coarse-Grained Mapping Algorithm for the Martini Force Field and Benchmarks for Membrane–Water Partitioning”. In: *J. Chem. Theory Comput.* vol. 17, no. 9 (2021), pp. 5777–5791.

[20] Tetko, I. V., Tanchuk, V. Y., and Villa, A. E. P. “Prediction of n-Octanol/Water Partition Coefficients from PHYSPROP Database Using Artificial Neural Networks and E-State Indices”. In: *J. Chem. Inf. Comput. Sci.* vol. 41, no. 5 (2001), pp. 1407–1421.

[21] Tetko, I. V. and Tanchuk, V. Y. “Application of Associative Neural Networks for Prediction of Lipophilicity in ALOGPS 2.1 Program”. In: *J. Chem. Inf. Comput. Sci.* vol. 42, no. 5 (2002), pp. 1136–1145.

[22] Rühle, V. et al. “Versatile Object-Oriented Toolkit for Coarse-Graining Applications”. In: *J. Chem. Theory Comput.* vol. 5, no. 12 (2009), pp. 3211–3223.

[23] Mirzoev, A. and Lyubartsev, A. P. “MagiC: Software Package for Multiscale Modeling”. In: *J. Chem. Theory Comput.* vol. 9, no. 3 (2013), pp. 1512–1520.

[24] Bejagam, K. K. et al. “Machine-Learned Coarse-Grained Models”. In: *J. Phys. Chem. Lett.* vol. 9, no. 16 (2018), pp. 4667–4672.

[25] Bejagam, K. K. et al. “PSO-Assisted Development of New Transferable Coarse-Grained Water Models”. In: *J. Phys. Chem. B* vol. 122, no. 6 (2018), pp. 1958–1971.

[26] Taghiyeh, S. and Xu, J. “A new particle swarm optimization algorithm for noisy optimization problems”. In: *Swarm Intell* vol. 10, no. 3 (2016), pp. 161–192.

[27] Meer, G. van, Voelker, D. R., and Feigenson, G. W. “Membrane lipids: where they are and how they behave”. In: *Nat. Rev. Mol. Cell Biol.* vol. 9, no. 2 (2008), pp. 112–124.

[28] Eberhart, R. and Kennedy, J. “A new optimizer using particle swarm theory”. In: *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995.

[29] Wang, F., Zhang, H., and Zhou, A. “A particle swarm optimization algorithm for mixed-variable optimization problems”. In: *Swarm Evol. Comput.* vol. 60 (2021), p. 100808.

[30] Rubner, Y., Tomasi, C., and Guibas, L. J. “The Earth Mover’s Distance as a Metric for Image Retrieval”. In: *Int J Comput Vision* vol. 40, no. 2 (2000), pp. 99–121.

[31] Rada-Vilela, J., Johnston, M., and Zhang, M. “Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems”. In: *Swarm Evol. Comput.* vol. 17 (2014), pp. 37–59.

[32] Pan, H., Wang, L., and Liu, B. “Particle swarm optimization for function optimization in noisy environment”. In: *Appl Math Comput* vol. 181, no. 2 (2006), pp. 908–919.

[33] Chen, C.-H. et al. “Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization”. In: *Discrete Event Dyn Syst* vol. 10, no. 3 (2000), pp. 251–270.

[34] Boesel, J., Nelson, B. L., and Kim, S.-H. “Using Ranking and Selection to "Clean up" after Simulation Optimization”. In: *Oper. Res.* vol. 51, no. 5 (2003), pp. 814–825.

[35] Methorst, J., Hilten, N. van, and Risselada, H. J. “Inverse design of cholesterol attracting transmembrane helices reveals a paradoxical role of hydrophobic length”. In: *bioRxiv* (2021).

[36] Abraham, M. J. et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”. In: *SoftwareX* vol. 1-2 (2015), pp. 19–25.

[37] Michaud-Agrawal, N. et al. “MDAnalysis: A toolkit for the analysis of molecular dynamics simulations”. In: *J. Comput. Chem.* vol. 32, no. 10 (2011), pp. 2319–2327.

[38] Gowers, R. et al. “MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations”. In: *Proceedings of the 15th Python in Science Conference*. SciPy, 2016.

[39] Smith, P. and Lorenz, C. D. “LiPyphilic: A Python Toolkit for the Analysis of Lipid Membrane Simulations”. In: *J. Chem. Theory Comput.* vol. 17, no. 9 (2021), pp. 5907–5919.

[40] Virtanen, P. et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nat. Methods* vol. 17, no. 3 (2020), pp. 261–272.

[41] Pele, O. and Werman, M. “Fast and robust Earth Mover’s Distances”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009.

[42] Pele, O. and Werman, M. “A Linear Time Histogram Metric for Improved SIFT Matching”. In: *Computer Vision – ECCV 2008*. Ed. by Forsyth, D., Torr, P., and Zisserman, A. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 495–508.

[43] Nguyen, H., Case, D. A., and Rose, A. S. “NGLview–interactive molecular graphics for Jupyter notebooks”. In: *Bioinformatics* vol. 34, no. 7 (2017). Ed. by Valencia, A., pp. 1241–1242.

[44] Klauda, J. B. et al. “Update of the CHARMM All-Atom Additive Force Field for Lipids: Validation on Six Lipid Types”. In: *J. Phys. Chem. B* vol. 114, no. 23 (2010), pp. 7830–7843.

[45] Venable, R. M. et al. “CHARMM All-Atom Additive Force Field for Sphingomyelin: Elucidation of Hydrogen Bonding and of Positive Curvature”. In: *Biophys. J.* vol. 107, no. 1 (2014), pp. 134–145.

[46] Wang, E. and Klauda, J. B. “Molecular Dynamics Simulations of Ceramide and Ceramide-Phosphatidylcholine Bilayers”. In: *J. Phys. Chem. B* vol. 121, no. 43 (2017), pp. 10091–10104.

[47] Jo, S. et al. “CHARMM-GUI Membrane Builder for Mixed Bilayers and Its Application to Yeast Membranes”. In: *Biophys. J.* vol. 97, no. 1 (2009), pp. 50–58.

[48] Wu, E. L. et al. “CHARMM-GUI Membrane Builder toward realistic biological membrane simulations”. In: *J. Comput. Chem.* vol. 35, no. 27 (2014), pp. 1997–2004.

[49] Lee, J. et al. “CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field”. In: *J. Chem. Theory Comput.* vol. 12, no. 1 (2015), pp. 405–413.

[50] Hess, B. et al. “LINCS: A linear constraint solver for molecular simulations”. In: *J. Comput. Chem.* vol. 18, no. 12 (1997), pp. 1463–1472. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291096-987X%28199709%2918%3A12%3C1463%3A%3AAID-JCC4%3E3.0.CO%3B2-H.

[51] Darden, T., York, D., and Pedersen, L. “Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems”. In: *J. Chem. Phys.* vol. 98, no. 12 (1993), pp. 10089–10092.

[52] Bussi, G., Donadio, D., and Parrinello, M. “Canonical sampling through velocity rescaling”. In: *J. Chem. Phys.* vol. 126, no. 1 (2007), p. 014101.

[53] Parrinello, M. and Rahman, A. “Polymorphic transitions in single crystals: A new molecular dynamics method”. In: *J Appl Phys* vol. 52, no. 12 (1981), pp. 7182–7190.

[54] Borges-Araújo, L. et al. “Martini 3 Coarse-Grained Force Field for Cholesterol”. In: *J. Chem. Theory Comput.* (2023).

[55] Borges-Araújo, L. et al. *Parameterization of cholesterol for the Martini 3 coarse grained force field*. 2023. URL: https://github.com/Martini-Force-Field-Initiative/M3-Sterol-Parameters (visited on 06/09/2023).

[56] Wassenaar, T. A. et al. “Computational Lipidomics with insane: A Versatile Tool for Generating Custom Membranes for Molecular Simulations”. In: *J. Chem. Theory Comput.* vol. 11, no. 5 (2015), pp. 2144–2155.

[57] Berendsen, H. J. C. et al. “Molecular dynamics with coupling to an external bath”. In: *J. Chem. Phys.* vol. 81, no. 8 (1984), pp. 3684–3690.

[58] Kowalik, B. et al. “Combination of MD Simulations with Two-State Kinetic Rate Modeling Elucidates the Chain Melting Transition of Phospholipid Bilayers for Different Hydration Levels”. In: *J. Phys. Chem. B* vol. 119, no. 44 (2015), pp. 14157–14167.

[59] Sun, L. and Böckmann, R. A. “Membrane phase transition during heating and cooling: molecular insight into reversible melting”. In: *Eur. Biophys. J.* vol. 47, no. 2 (2017), pp. 151–164.

[60] Coppock, P. S. and Kindt, J. T. “Determination of Phase Transition Temperatures for Atomistic Models of Lipids from Temperature-Dependent Stripe Domain Growth Kinetics”. In: *J. Phys. Chem. B* vol. 114, no. 35 (2010), pp. 11468–11473.

[61] Borges-Araújo, L. et al. “Improved Parameterization of Phosphatidylinositide Lipid Headgroups for the Martini 3 Coarse-Grain Force Field”. In: *J. Chem. Theory Comput.* vol. 18, no. 1 (2021), pp. 357–373.

[62] Rada-Vilela, J., Johnston, M., and Zhang, M. “Deception, blindness and disorientation in particle swarm optimization applied to noisy problems”. In: *Swarm Intell* vol. 8, no. 4 (2014), pp. 247–273.

[63] Keyvanloo, A. et al. “The Phase Behavior and Organization of Sphingomyelin/Cholesterol Membranes: A Deuterium NMR Study”. In: *Biophys. J.* vol. 114, no. 6 (2018), pp. 1344–1356.

[64] Marrink, S. J., Risselada, J., and Mark, A. E. “Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model”. In: *Chem. Phys. Lipids* vol. 135, no. 2 (2005), pp. 223–244.

[65] Nobile, M. S. et al. “Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization”. In: *Swarm Evol. Comput.* vol. 39 (2018), pp. 70–85.

[66] Rada-Vilela, J., Johnston, M., and Zhang, M. “Population statistics for particle swarm optimization: Hybrid methods in noisy optimization problems”. In: *Swarm Evol. Comput.* vol. 22 (2015), pp. 15–29.

[67] Melo, M. N., Ingólfsson, H. I., and Marrink, S. J. “Parameters for Martini sterols and hopanoids based on a virtual-site description”. In: *J. Chem. Phys.* vol. 143, no. 24 (2015).

[68] Alessandri, R. et al. “Martini 3 Coarse-Grained Force Field: Small Molecules”. In: *Adv. Theory Simul.* vol. 5, no. 1 (2021), p. 2100391.

[69] Alessandri, R. et al. “A Practical Introduction to Martini 3 and its Application to Protein-Ligand Binding Simulations”. In: *A Practical Guide to Recent Advances in Multiscale Modeling and Simulation of Biomolecules*. AIP Publishing LLCMelville, New York, 2023, pp. 1–1–1–34.

[70] Uusitalo, J. J. et al. “Martini Coarse-Grained Force Field: Extension to DNA”. In: *J. Chem. Theory Comput.* vol. 11, no. 8 (2015), pp. 3932–3945.

[71] Vazquez-Salazar, L. I. et al. “Martini coarse-grained models of imidazolium-based ionic liquids: from nanostructural organization to liquid–liquid extraction”. In: *Green Chem.* vol. 22, no. 21 (2020), pp. 7376–7386.

[72] Bulacu, M. et al. “Improved Angle Potentials for Coarse-Grained Molecular Dynamics Simulations”. In: *J. Chem. Theory Comput.* vol. 9, no. 8 (2013), pp. 3282–3292.

[73] Fábián, B., Thallmair, S., and Hummer, G. “Optimal Bond Constraint Topology for Molecular Dynamics Simulations of Cholesterol”. In: *J. Chem. Theory Comput.* vol. 19, no. 5 (2023), pp. 1592–1601.

[74] Thallmair, S. et al. “Nonconverged Constraints Cause Artificial Temperature Gradients in Lipid Bilayer Simulations”. In: *J. Phys. Chem. B* vol. 125, no. 33 (2021), pp. 9537–9546.

[75] Norgaard, A. B., Ferkinghoff-Borg, J., and Lindorff-Larsen, K. “Experimental Parameterization of an Energy Function for the Simulation of Unfolded Proteins”. In: *Biophys. J.* vol. 94, no. 1 (2008), pp. 182–192.

[76] Li, D.-W. and Brüschweiler, R. “Iterative Optimization of Molecular Mechanics Force Fields from NMR Data of Full-Length Proteins”. In: *J. Chem. Theory Comput.* vol. 7, no. 6 (2011), pp. 1773–1782.

[77] Carmichael, S. P. and Shell, M. S. “A New Multiscale Algorithm and Its Application to Coarse-Grained Peptide Models for Self-Assembly”. In: *J. Phys. Chem. B* vol. 116, no. 29 (2012), pp. 8383–8393.

[78] Gad, A. G. “Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review”. In: *Arch Comput Method E* vol. 29, no. 5 (2022), pp. 2531–2561.

[79] Carpenter, T. S. et al. “Capturing Phase Behavior of Ternary Lipid Mixtures with a Refined Martini Coarse-Grained Force Field”. In: *J. Chem. Theory Comput.* vol. 14, no. 11 (2018), pp. 6050–6062.

[80] Akaike, H. “Information Theory and an Extension of the Maximum Likelihood Principle”. In: *Proc. of the Second Internat. Symp. on Information Theory, edited by B. N. Petrov and S. Caski.* (1973), pp. 267–281.

[81] Akaike, H. “A new look at the statistical model identification”. In: *IEEE Trans. Automat. Control* vol. 19, no. 6 (1974), pp. 716–723.

[82] Hurvich, C. M. and Tsai, C.-L. “Regression and time series model selection in small samples”. In: *Biometrika* vol. 76, no. 2 (1989), pp. 297–307.

Chapter 4

Summary and general discussion

Both lines of work presented in this thesis are related to sampling mathematical spaces in a biomolecular context, i.e., phase space (curvature sensing) and parameter space (molecule parameterization).

4.1 Curvature / packing defect sensing

There is substantial interest in curvature sensing proteins, due to their involvement in a large variety of biological processes [25, 26]. It is therefore rather surprising that methods to explore curvature-dependent free-energy landscapes are quite rare.

In Chapter 2, a method to quantify membrane curvature sensing of peripheral membrane proteins was developed. Through umbrella sampling along a buckled membrane, this method allows the generation of curvature-dependent free energy profiles. When moving along the buckled membrane, curvature is sampled in a continuous fashion from the minimum curvature $K_{\mathrm{min}} = -1/R_{\mathrm{min}}$ to the maximum curvature $K_{\mathrm{max}} = 1/R_{\mathrm{min}}$, where the minimal radius R_{min} can assume extreme values of $\approx 4\,\mathrm{nm}$. Thus, the relative free energy can be estimated over a very broad curvature range. Albeit, the buckled geometry comes with some limitations. i) The protein can only have small binding domains. Otherwise, the protein covers a broad curvature range, i.e., the position in curvature space is ill-defined. ii) For proteins that extend far away from the membrane, steric clashes in negatively curved region cannot be avoided.

Buckled membranes in conjunction with unbiased MD simulations were used before to study curvature sensing of proteins [27, 28], but this is of limited use in a quantitative approach. Without a biasing potential, sampling is limited to states close to the system's minimum free energy. Obviously, for a curvature sensing protein, this means only a limited curvature range will be sampled. A less obvious consequence comes to pass when the protein's preferred curvature K_{p} lies outside the buckled membrane's curvature range. Assuming $K_{\mathrm{p}} \geq K_{\mathrm{max}}$, then the protein will mostly sample the top of the buckle, since this is the free-energy minimum of the system. As the membrane shape description is based on the arc length parameter s in all of these methods, the free energy as a function of curvature is given by $F(K) = F(s) + k_{\mathrm{B}}T \ln(|dK/ds|)$, and $dK/ds \to 0$ at the top of the buckle. Therefore, the transformation to the desired quantity $F(K)$ breaks down in the region that is sampled the most.

Curvature sensing quantification was showcased using two known curvature sensing peptides (ALPS and α-synuclein), which have contrasting chemical properties and curvature sensing mechanisms. The estimated intrinsic curvature of α-synuclein matched findings from tubulation experiments [29].

The method was employed to investigate the curvature sensing ability of the lipid transport protein Ups1 in Paper 2 [11]. The protein complex comprised of Ups1 and Mdm35 is believed to shuttle phosphatidic acid (PA) from the outer to the inner membrane of mitochondria [30]. Ups1 binds to membranes with the help of a hydrophobic loop [31, 32]. Similarly to amphipathic helices, such loops can act as membrane curvature sensors [25]. The umbrella sampling and buckling method was used to resolve the relative free energy of binding to a membrane with a lipid composition resembling that of mitochondrial membranes. Preferential binding to positively curved regions of the membrane was found. This result was confirmed via a co-flotation assay, which probed binding to differently sized vesicles. Additionally, the buckled membrane was used to estimate relative free energies of lipid extraction. It was found that the work required to extract a PA lipid is lowered for positively curved regions and increased for negatively curved regions.

In Paper 3 [10], the relationship of curvature sensing with lipid packing defect sensing was exploited further. In this work, membrane defects were generated by applying tension to the membrane. Additionally, an end state free energy method was developed that allows for rapid comparison of the binding free energies of peptides. Its usefulness lies therefore more in quick estimation in high-throughput approaches, while the buckling method provides a detailed curvature-dependent free energy profile.

4.2 Automated coarse-grained molecule parameterization

Automation in simulation is becoming more and more important [33]. As applications become wider and more complex, automation seems to be the only way to handle this. There is a broad range of automated set-up and analysis tools – e.g., CHARMM-GUI [34–37], MemProtMD [38, 39], Polyply [40] – and automation in FF development and parameterization [41–43], is one manifestation of this trend. The desire from basic research and technological applications to replicate nature's complexity, makes the need for a growing number of parameterized molecules, and therefore automation, even more pressing [44, 45]. The broader coverage of chemical space in Martini 3 opened up new possibilities [21], but has concomitantly made parameterization less intuitive. A problem that can be alleviated with standardized parameterization pipelines.

Former automated parameterization methods focused on bonded interactions only [43, 46]. Others that addressed mapping, bonded and non-bonded parameters were solely focused on small molecules, and simple target objectives [41, 42]. The parameterization method presented in Chapter 3 is tailor-made for parameterization tasks in building-block force fields, such as Martini 3. The mixed-variable PSO approach is particularly suited for force fields, where non-bonded interactions are specified by a fixed set of bead types. Parameterization of bonded and non-bonded parameters is not independent, rather they are connected via the density of interactions [21, 23, 24], as shown in Section 1.2. While the demonstration of the method was performed on a lipid and focused strongly on optimization of targets where Martini lipids have not performed well in the past, the method is applicable for any kind of molecule. The method is applicable in high-throughput setups as well. Training systems and observables are easily modified for a specific parameterization task.

For high-throughput applications, inclusion of an automated mapping scheme would be desirable. When parameterizing a molecule in the traditional non-automated fashion, mapping, and the choice of bonded and non-bonded parameters are two distinct steps. The existing automated tools have treated this problem in the same way. When mapping and parameter choice are addressed individually, an automated mapping scheme would simply be a step in the parameterization pipeline that happens before the mixed-variable particle swarm optimization (mv-PSO) algorithm, and mv-PSO can be used without any adaptation. Automated mapping for small molecules had been done for Martini 2, but due to the changes in Martini 3 this problem needs to be revisited [45]. A promising way of going about this task seems to be the implementation of small molecule design rules proposed by Alessandri et al. [47].

A much more complex task is an adaptive mapping that is part of the optimization. As the number of possibilities explodes quickly with molecule size, this seems feasible only for small parts of a molecule. An example would be a change of resolution, e.g., using one regular bead (4-1 mapping) or two tiny beads (2-1 mapping). How to overcome the concomitant change of PSO search space dimensionality is an open problem.

While Martini lipid models do capture important membrane phase behavior, such as the gel to fluid phase transition [48], and liquid-ordered liquid-disordered phase separation [49], they do so rather inaccurately. Melting temperatures are tens of K too low (cf. ref. [48] and Chapter 3). Liquid-ordered liquid-disordered phase separation, which is generally observed in ternary mixtures consisting of a high-T_m lipid species (e.g. DPPC, sphingomyelin), a low-T_m lipid species (e.g. DLiPC, DOPC), and cholesterol, is captured in Martini only with polyunsaturated lipids (e.g. DLiPC), but not monounsaturated lipids (e.g. DOPC) [50]. Such inaccuracies are not entirely unexpected, given that these phenomena were not targeted in the original parameterization [20]. Given the need for higher fidelity models, phase behavior should be targeted in a reparameterization of Martini lipids. While a recent reparameterization of the Martini 2 tails claimed to capture the phase behavior of ternary lipid mixtures [51], the phase separation was actually a result of non-converged constraints in the CHOL model causing a temperature gradient in the bilayer [52, 53].

As membrane phase behavior is based on collective processes, accurately measuring the corresponding observables, such as melting temperature or enrichment index [54], which can be used to gauge phase separation in ternary mixtures, usually requires a substantial computational effort. Therefore, their usage in an automated parameterization tool is no easy feat. The challenge lies in finding a measurement procedure that is sufficiently efficient and accurate. In a multi-objective optimization problem, which is what molecule parameterization is, large inaccuracies of one observable will not only hamper optimization of the corresponding objective, but deteriorate overall performance. In this regard, the usage of noise-mitigation methods can help. As shown in Chapter 3, focused resampling of observables with large variances for good solutions, lead to overall better sphingomyelin models.

Reparameterization of the sphingolipid linker region with T_m as one of several target observables gave satisfactory results, cf. Chapter 3. Instead of T_m being $\approx 20\,\mathrm{K}$ too low, as was the case for the old model, T_m of the best candidate solutions is within $\approx 2\,\mathrm{K}$ of the experimental value, which corresponds to the experimental and computational uncertainties. Other important targets, such as bond length and angle distributions, area per lipid and membrane thickness, were matched as well.

The parameterization shown in Chapter 3 did not explicitly include phase behavior of ternary mixtures as an objective. In principle, the algorithm is capable of generating solutions, that reproduce liquid-ordered liquid-disordered phase separation in a DOPC/DPSM/CHOL mixture (cf. Figure 4.1), as established in experiments [55], but matching phase separation seemed incompatible with

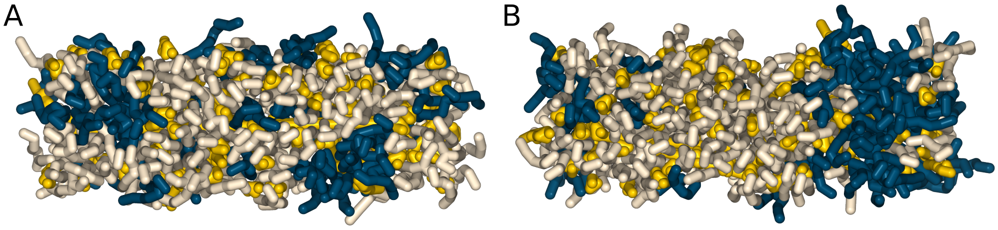


Figure 4.1: Phase behavior of ternary mixture with 22 % DOPC (blue) 48 % DPSM (white) 30 % CHOL (yellow). **A** old DPSM **B** DPSM model optimized solely on reproducing correct phase separation behavior.

reproduction of other target observables. Phase behavior of lipid membranes is largely influenced by tail-tail interactions. When only reparameterizing the linker region, it seems therefore likely that the linker parameters have to compensate for the lack of accuracy in the tail parameters, at the cost of lowered quality regarding other objectives. A reparameterization of other lipid building blocks, particularly the acyl chains, appears to be required for an overall improved model. A task for which mv-PSO is the ideal tool, as parameterization of one building block can be done simultaneously for different molecules that share this building block, e.g., a linker region, with different head groups and tails, or tails with several lipid classes.

In general, tools such as mv-PSO are perfectly suited for large collaborative parameterization efforts and the population of databases such as the Martini database MAD [56], as this allows for a clear definition and reproducible implementation of a force field's parameterization philosophy, i.e., choice of thermodynamic state points, target observables, target data sources, training systems, and importance of different observables. Consistency in parameterization can be beneficial to transferability [41, 57].

Automated parameterization with a preset pipeline could also be applied in novelty design of molecules, such as functional lipids for soft nanoparticles (drug-delivery, etc. [45]). An expert with domain knowledge makes chemical adjustments, and the mv-PSO algorithm provides an automated, standardized parameterization pipeline, that ensures good reproduction of important targets.

An alternative would be a physics-based inverse-design approach, where a CG model is optimized for a certain behavior. Optimizing bead types and charge (equivalent to changing functional groups) is already possible with the current implementation. A flexible mapping scheme, i.e., the ability to add beads and bonds, would open up even more possibilities. Also combinations with a genetic algorithm, similar to ref. [58], are imaginable. Notably, any physics-based inverse design strategy would strongly profit from an improved quality of CG models, as such design strategies rely heavily on the relevant physics being embedded in the FF and molecule parameterizations.

Bibliography

[1] Bagatolli, L. and Mouritsen, O. G. *Life as a matter of fat: lipids in a membrane biophysics perpective (second edition); Ole G. Mouritsen and Luis A. Bagatolli*. 2015.

[2] Harayama, T. and Riezman, H. “Understanding the diversity of membrane lipid composition”. In: *Nat. Rev. Mol. Cell Biol.* vol. 19, no. 5 (2018), pp. 281–296.

[3] Klose, C., Surma, M. A., and Simons, K. “Organellar lipidomics—background and perspectives”. In: *Curr. Opin. Cell Biol.* vol. 25, no. 4 (2013), pp. 406–413.

[4] Meer, G. van. “Cellular lipidomics”. In: *The EMBO Journal* vol. 24, no. 18 (2005), pp. 3159–3165.

[5] Rawicz, W. et al. “Effect of Chain Length and Unsaturation on Elasticity of Lipid Bilayers”. In: *Biophys. J.* vol. 79, no. 1 (2000), pp. 328–339.

[6] Ballweg, S. et al. “Regulation of lipid saturation without sensing membrane fluidity”. In: *Nat. Commun.* vol. 11, no. 1 (2020).

[7] Poojari, C. S., Scherer, K. C., and Hub, J. S. “Free energies of membrane stalk formation from a lipidomics perspective”. In: *Nat. Commun.* vol. 12, no. 1 (2021).

[8] Peter, B. J. “BAR Domains as Sensors of Membrane Curvature: The Amphiphysin BAR Structure”. In: *Science* vol. 303, no. 5657 (2004), pp. 495–499.

[9] Hilten, N. van, Stroh, K. S., and Risselada, H. J. “Membrane Thinning Induces Sorting of Lipids and the Amphipathic Lipid Packing Sensor (ALPS) Protein Motif”. In: *Front. Physiol.* vol. 11 (2020).

[10] Hilten, N. van, Stroh, K. S., and Risselada, H. J. “Efficient Quantification of Lipid Packing Defect Sensing by Amphipathic Peptides: Comparing Martini 2 and 3 with CHARMM36”. In: *J. Chem. Theory Comput.* vol. 18, no. 7 (2022), pp. 4503–4514.

[11] Sadeqi, F. et al. “Membrane interactions of mitochondrial lipid transfer proteins”. In: *bioRxiv* (2022).

[12] Mehner-Breitfeld, D. et al. “TatA and TatB generate a hydrophobic mismatch important for the function and assembly of the Tat translocon in Escherichia coli”. In: *J. Biol. Chem.* vol. 298, no. 9 (2022), p. 102236.

[13] Marrink, S. J. et al. “Computational Modeling of Realistic Cell Membranes”. In: *Chem. Rev.* vol. 119, no. 9 (2019), pp. 6184–6226.

[14] Corradi, V. et al. “Emerging Diversity in Lipid–Protein Interactions”. In: *Chem. Rev.* vol. 119, no. 9 (2019), pp. 5775–5848.

[15] Muller, M. P. et al. “Characterization of Lipid–Protein Interactions and Lipid-Mediated Modulation of Membrane Protein Function through Molecular Simulation”. In: *Chem. Rev.* vol. 119, no. 9 (2019), pp. 6086–6161.

[16] Leonard, A. N. et al. “Developing and Testing of Lipid Force Fields with Applications to Modeling Cellular Membranes”. In: *Chem. Rev.* vol. 119, no. 9 (2019), pp. 6227–6269.

[17] Enkavi, G. et al. “Multiscale Simulations of Biological Membranes: The Challenge To Understand Biological Phenomena in a Living Substance”. In: *Chem. Rev.* vol. 119, no. 9 (2019), pp. 5607–5774.

[18] Meer, G. van, Voelker, D. R., and Feigenson, G. W. “Membrane lipids: where they are and how they behave”. In: *Nat. Rev. Mol. Cell Biol.* vol. 9, no. 2 (2008), pp. 112–124.

[19] Ingólfsson, H. I. et al. “Computational ‘microscopy’ of cellular membranes”. In: *J. Cell Sci.* (2016).

[20] Marrink, S. J. et al. “The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations”. In: *The Journal of Physical Chemistry B* vol. 111, no. 27 (2007), pp. 7812–7824.

[21] Souza, P. C. T. et al. “Martini 3: a general purpose force field for coarse-grained molecular dynamics”. In: *Nat. Methods* vol. 18, no. 4 (2021), pp. 382–388.

[22] Bradley, R. and Radhakrishnan, R. “Coarse-Grained Models for Protein-Cell Membrane Interactions”. In: *Polymers* vol. 5, no. 3 (2013), pp. 890–936.

[23] Alessandri, R. et al. “Pitfalls of the Martini Model”. In: *J. Chem. Theory Comput.* vol. 15, no. 10 (2019), pp. 5448–5460.

[24] Risselada, H. J. “Martini 3: a coarse-grained force field with an eye for atomic detail”. In: *Nat. Methods* vol. 18, no. 4 (2021), pp. 342–343.

[25] Antonny, B. “Mechanisms of Membrane Curvature Sensing”. In: *Annu. Rev. Biochem.* vol. 80, no. 1 (2011), pp. 101–123.

[26] Baumgart, T. et al. “Thermodynamics and Mechanics of Membrane Curvature Generation and Sensing by Proteins and Lipids”. In: *Annu. Rev. Phys. Chem.* vol. 62, no. 1 (2011), pp. 483–506.

[27] Gómez-Llobregat, J., Elías-Wolff, F., and Lindén, M. “Anisotropic Membrane Curvature Sensing by Amphipathic Peptides”. In: *Biophys. J.* vol. 110, no. 1 (2016), pp. 197–204.

[28] Martyna, A. et al. “Curvature Sensing by a Viral Scission Protein”. In: *Biochemistry* vol. 55, no. 25 (2016), pp. 3493–3496.

[29] Varkey, J. et al. “Membrane Curvature Induction and Tubulation Are Common Features of Synucleins and Apolipoproteins”. In: *J. Biol. Chem.* vol. 285, no. 42 (2010), pp. 32486–32493.

[30] Tamura, Y., Kawano, S., and Endo, T. “Lipid homeostasis in mitochondria”. In: *Biol. Chem.* vol. 401, no. 6-7 (2020), pp. 821–833.

[31] Miliara, X. et al. “Structural determinants of lipid specificity within Ups/PRELI lipid transfer proteins”. In: *Nat. Commun.* vol. 10, no. 1 (2019).

[32] Lu, J. et al. “Molecular mechanism of mitochondrial phosphatidate transfer by Ups1”. In: *Commun. Biol.* vol. 3, no. 1 (2020).

[33] Bereau, T. “Computational compound screening of biomolecules and soft materials by molecular simulations”. In: *Model Simul Mater Sc* vol. 29, no. 2 (2021), p. 023001.

[34] Jo, S. et al. “CHARMM-GUI: A web-based graphical user interface for CHARMM”. In: *J. Comput. Chem.* vol. 29, no. 11 (2008), pp. 1859–1865.

[35] Jo, S. et al. “CHARMM-GUI Membrane Builder for Mixed Bilayers and Its Application to Yeast Membranes”. In: *Biophys. J.* vol. 97, no. 1 (2009), pp. 50–58.

[36] Jo, S., Kim, T., and Im, W. “Automated Builder and Database of Protein/Membrane Complexes for Molecular Dynamics Simulations”. In: *PLoS ONE* vol. 2, no. 9 (2007). Ed. by Yuan, A., e880.

[37] Wu, E. L. et al. “CHARMM-GUI Membrane Builder toward realistic biological membrane simulations”. In: *J. Comput. Chem.* vol. 35, no. 27 (2014), pp. 1997–2004.

[38] Stansfeld, P. J. et al. “MemProtMD: Automated Insertion of Membrane Protein Structures into Explicit Lipid Membranes”. In: *Structure* vol. 23, no. 7 (2015), pp. 1350–1361.

[39] Newport, T. D., Sansom, M. S. P., and Stansfeld, P. J. "The MemProtMD database: a resource for membrane-embedded protein structures and their lipid interactions". In: *Nucleic Acids Res.* vol. 47, no. D1 (2018), pp. D390–D397.

[40] Grünewald, F. et al. "Polyply; a python suite for facilitating simulations of macromolecules and nanomaterials". In: *Nat. Commun.* vol. 13, no. 1 (2022).

[41] Bereau, T. and Kremer, K. "Automated Parametrization of the Coarse-Grained Martini Force Field for Small Organic Molecules". In: *J. Chem. Theory Comput.* vol. 11, no. 6 (2015), pp. 2783–2791.

[42] Potter, T. D., Barrett, E. L., and Miller, M. A. "Automated Coarse-Grained Mapping Algorithm for the Martini Force Field and Benchmarks for Membrane–Water Partitioning". In: *J. Chem. Theory Comput.* vol. 17, no. 9 (2021), pp. 5777–5791.

[43] Empereur-Mot, C. et al. "Swarm-CG: Automatic Parametrization of Bonded Terms in MARTINI-Based Coarse-Grained Models of Simple to Complex Molecules via Fuzzy Self-Tuning Particle Swarm Optimization". In: *ACS Omega* vol. 5, no. 50 (2020), pp. 32823–32843.

[44] Marrink, S. J. et al. "Two decades of Martini: Better beads, broader scope". In: *WIREs Computational Molecular Science* (2022).

[45] Kjølbye, L. R. et al. "Towards design of drugs and delivery systems with the Martini coarse-grained model". In: *QRB Discovery* vol. 3 (2022).

[46] Graham, J. A., Essex, J. W., and Khalid, S. "PyCGTOOL: Automated Generation of Coarse-Grained Molecular Dynamics Models from Atomistic Trajectories". In: *J. Chem. Inf. Model.* vol. 57, no. 4 (2017), pp. 650–656.

[47] Alessandri, R. et al. "Martini 3 Coarse-Grained Force Field: Small Molecules". In: *Adv. Theory Simul.* vol. 5, no. 1 (2021), p. 2100391.

[48] Marrink, S. J., Risselada, J., and Mark, A. E. "Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model". In: *Chem. Phys. Lipids* vol. 135, no. 2 (2005), pp. 223–244.

[49] Risselada, H. J. and Marrink, S. J. "The molecular face of lipid rafts in model membranes". In: *Proc. Natl. Acad. Sci.* vol. 105, no. 45 (2008), pp. 17367–17372.

[50] Thallmair, S., Ingólfsson, H. I., and Marrink, S. J. "Cholesterol Flip-Flop Impacts Domain Registration in Plasma Membrane Models". In: *J. Phys. Chem. Lett.* vol. 9, no. 18 (2018), pp. 5527–5533.

[51] Carpenter, T. S. et al. “Capturing Phase Behavior of Ternary Lipid Mixtures with a Refined Martini Coarse-Grained Force Field”. In: *J. Chem. Theory Comput.* vol. 14, no. 11 (2018), pp. 6050–6062.

[52] Javanainen, M., Fabian, B., and Martinez-Seara, H. *Comment on "Capturing Phase Behavior of Ternary Lipid Mixtures with a Refined Martini Coarse-Grained Force Field"*. 2020.

[53] Thallmair, S. et al. “Nonconverged Constraints Cause Artificial Temperature Gradients in Lipid Bilayer Simulations”. In: *J. Phys. Chem. B* vol. 125, no. 33 (2021), pp. 9537–9546.

[54] Smith, P. and Lorenz, C. D. “LiPyphilic: A Python Toolkit for the Analysis of Lipid Membrane Simulations”. In: *J. Chem. Theory Comput.* vol. 17, no. 9 (2021), pp. 5907–5919.

[55] Veatch, S. L. and Keller, S. L. “Organization in Lipid Membranes Containing Cholesterol”. In: *Phys. Rev. Lett.* vol. 89, no. 26 (2002), p. 268101.

[56] Hilpert, C. et al. “Facilitating CG simulations with MAD: the MArtini Database Server”. In: *bioRxiv* (2022).

[57] Vanommeslaeghe, K. et al. “CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields”. In: *J. Comput. Chem.* (2009), NA–NA.

[58] Methorst, J., Hilten, N. van, and Risselada, H. J. “Inverse design of cholesterol attracting transmembrane helices reveals a paradoxical role of hydrophobic length”. In: *bioRxiv* (2021).

Acknowledgements

Firstly, I would like to express my deep gratitude to my supervisor Jelger Risselada for giving me the opportunity to address interesting and challenging scientific topics, and for allowing me to pursue my own ideas.

I am extremely grateful to Marcus Müller for many insightful comments and discussions, from which I learned a lot. Special thanks to my thesis advisory committee Stefan Klumpp and Matthias Krüger for their scientific guidance. I am also thankful to Michael Meinecke for the good collaboration in the curvature sensing project. Additionally, I would like to extend my sincere thanks to Luca Monticelli and Paulo C.T. Souza for sharing their immense knowledge and giving plentiful advice.

This endeavor would not have been possible without Yuliya Smirnova and Veronica Chappa. Our discussions about science and beyond, that could last for hours, were invaluable.

Thanks should also go to Tabea Oswald, Katrin Glormann and Gabriele Schuster for navigating through bureaucratic obstacles.

Many thanks to all my fellow PhD students, who made my time in Göttingen memorable.

Laura, it was great to have you as a colleague, I hope we manage to stay friends.

Margo and Magda, a huge thank you for friendship, support, and many conversations about the "bio"-side of science.

Finally, I thank my family for everything.

♥