Chapter 5

A Practical View of the Martini Force Field

Bart M. H. Bruininks, Paulo C. T. Souza, and Siewert J. Marrink

Abstract

Martini is a coarse-grained (CG) force field suitable for molecular dynamics (MD) simulations of (bio) molecular systems. It is based on mapping of two to four heavy atoms to one CG particle. The effective interactions between the CG particles are parametrized to reproduce partitioning free energies of small chemical compounds between polar and apolar phases. In this chapter, a summary of the key elements of this CG force field is presented, followed by an example of practical application: a lipoplex-membrane fusion experiment. Formulated as hands-on practice, this chapter contains guidelines to build CG models of important biological systems, such as asymmetric bilayers and double-stranded DNA. Finally, a series of notes containing useful information, limitations, and tips are described in the last section.

Key words Coarse-grained models, Martini force field, Molecular dynamics simulations, Biomolecular systems

1 Introduction

The initial Martini coarse-grained (CG) force field was developed in 2003 to study lipid membrane properties [1–3]. It allowed to investigate the behavior of large lipid aggregates at spatial and timescales unachievable to atomistic MD simulations, while retaining enough resolution and chemical specificity to give a microscopic and dynamic picture still unavailable in experiments. The Martini force field was shown to be capable to address a wide range of lipid-based processes such as vesicle self-assembly, vesicle fusion, lamellar to inverted hexagonal phase transition, and the formation of the gel- and liquid-order phases [1–6]. Over the years, the applicability of the force field has expanded to most common biomolecules such as proteins [7, 8], sugars [9, 10], nucleotides [11, 12], and some important cofactors [13], as well as many nonbiological molecules including synthetic polymers [14–20] and nanoparticles [17–19, 21]. Examples of Martini CG models are shown in Fig. 1a. A complete list can be found under "downloads" at cgmartini.nl. Noteworthy is the high compatibility of the

Massimiliano Bonomi and Carlo Camilloni (eds.), *Biomolecular Simulations: Methods and Protocols*, Methods in Molecular Biology, vol. 2022, https://doi.org/10.1007/978-1-4939-9608-7_5, © Springer Science+Business Media, LLC, part of Springer Nature 2019

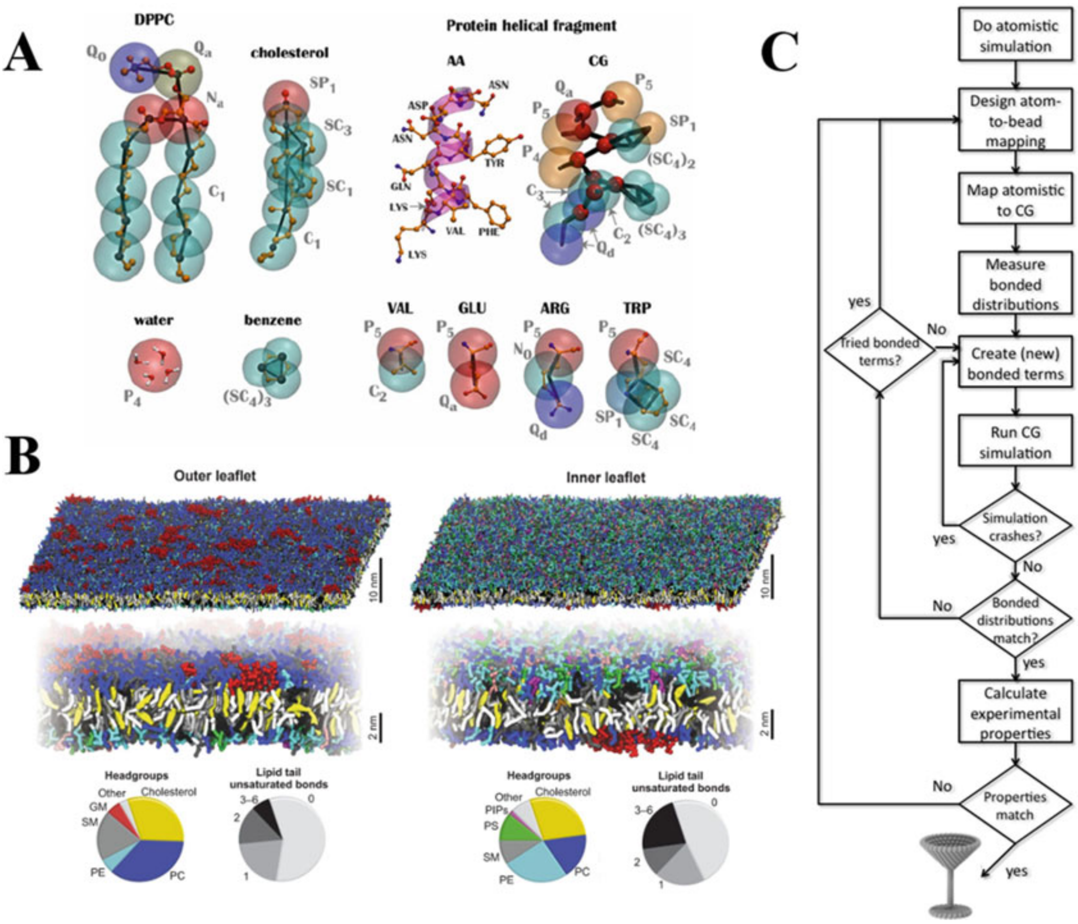

Fig. 1 Martini force field: (**a**) Some examples of Martini CG models used for lipids (DPPC and cholesterol), peptide, water, benzene, and some amino acids (adapted from [27]); (**b**) Example for a complex application: the idealized asymmetric plasma membrane comprises 63 different lipid types [22]; (**c**) Workflow for the parametrization of a new Martini CG model

individual models with each other. This allows for the modeling of complex biological environments such as the plasma membrane [22] (shown in Fig. 1b) and photosystem II in a thylakoid membrane [23].

This high compatibility is achieved by a clear modular mapping and parameterization scheme based on building blocks, called beads. Martini is a CG force field, which, in general, maps four nonhydrogen atoms to a single CG bead. During the mapping, chemical groups such as carboxylates or esters are represented by a single CG bead. This approach makes it easy to build new models based on the already available ones. The CG beads come in four chemical classes (or “flavors”): charged (Q), polar (P), nonpolar (N), and apolar (C). The Q and N classes each have four subtypes that are linked to their capability of participating in hydrogen bonding: donor and acceptor (da), donor (d), acceptor (a), and none (0). The main difference between these subtypes is their interaction strength with each other, allowing for a qualitative representation of hydrogen bonding. The P and C beads each have five subtypes, which represent a gradient from weak to strong polar or apolar properties, respectively. In total, this gives rise to 18 different bead chemical types. For computational efficiency, the mass of all standard beads is set to 72 amu, which equals the mass of four water molecules (represented by a P4 bead type in Martini).

Martini employs bonded and nonbonded potential forms commonly used in atomistic force fields, which make the model easy to be implemented in modern molecular dynamic programs as GROMACS [3, 5], GROMOS [24], and NAMD [25, 26]. Although this choice of potential forms is not the most accurate one for coarse-grained models (*see* **Notes 1** and **2**) [27], it enables Martini to take benefit of all the advances in high-performance parallel algorithms and enhanced sampling techniques developed in the past years. For the nonbonded interactions, 12–6 Lennard-Jones and Coulomb potentials are used (as shown in Eq. 1). In practice, these potentials are shifted and truncated for computational speedup. In the current implementation [5, 28], these potentials are both shifted such that the potentials reach 0 kJ/mol for any distance greater than 1.1 nm, the cutoff distance. In case of the LJ potential, ten levels of interaction are defined, differing in the LJ well depth (epsilon ranging from 5.6 to 2.0 kJ/mol), but with the same bead size (a sigma of 0.47 nm is used for the standard beads, except for interaction level IX, which has an increased sigma of 0.62 nm). For all possible pairs of CG bead types, one of those ten interaction levels has been assigned. These levels have been chosen based on the experimental water-oil partitioning of small molecules that are represented by each of the beads. Only the Q-beads bear an explicit charge and additionally interact via the Coulomb potential with a relative dielectric constant $\varepsilon_{\text{rel}} = 15$ for explicit screening. Together with the use of a shift function, this effectively results in a distance-dependent screening.

To allow for the mapping of aliphatic and aromatic ring structures (such as cyclohexane or benzene), a smaller bead (denoted with prefix "S") was introduced, mapping two or three nonhydrogen atoms to a single CG bead. The S beads have a reduced sigma of

$$V^{non\text{-}bonded} = \underbrace{\sum_{i,j} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]}_{V^{LJ}} + \underbrace{\sum_{i,j} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}}_{V^{Coul}}$$

Equation 1 The treatment of nonbonded interactions in the Martini force field is based in shifted and truncated 12–6 Lennard-Jones (V^{LJ}) and Coulomb potentials (V^{Coul})

0.43 nm and a scaled interaction strength corresponding to 75% of their standard bead counterpart. In the current model only S-S interactions make use of the reduced interaction scheme, that is, interaction with normal (N) particles are treated as N-N interactions. For the full parametrization scheme and interaction table, we would like to refer to the original paper of Marrink et al. [5] After the release of Martini 2.0 in 2007, an extra bead size has been added to the collection. The tiny (T) bead was introduced for the mapping of nucleotides and follows three/two-to-one mapping. Such a tiny bead is needed for correct stacking distance of the nucleobases in the double-stranded helix. The interaction strength of the T beads is not reduced, but their sigma is 0.32 nm. T beads interact with S and N beads as S-S and N-N, respectively (*see* **Note 3**) [11]. Besides the extra bead size, many new beads were introduced to satisfy the needs of specific models, as nucleic acids [11, 12], polymers [14–16, 20], nanoparticles [21], and some sugars [10, 29]. Polarizable water models [30, 31] have been designed for modeling of systems where implicit screening of electrostatic interactions caused by reorienting water dipoles is necessary (as discussed in **Note 4**).

Where the nonbonded interactions follow a "top-down" modeling approach (making use of experimental partitioning data), the bonded interactions are usually extracted via a "bottom-up" approach, based on reference atomistic data. This is achieved by mapping the heavy atoms of an atomistic simulation to a pseudo CG trajectory. The CG beads are usually placed at the centers of mass of the atoms they represent. From these pseudo CG trajectories, the bonded parameters can be extracted and compared to the CG model under development. By changing the bonded parameters, one should try to maximize the overlap of the conformational distributions of the pseudo CG and real CG models. This is an iterative process that should be repeated for achieving the best results. However, the philosophy of the Martini force field is to use (mainly) simple bonded potentials (as shown in Eq. 2); therefore, perfect overlap is not always achieved. Also, be aware that the bonded parameters might influence the partitioning of your molecule or even other macroscopic properties (e.g., area per lipid—APL—for bilayers and radius of gyration for polymers). Therefore, validation of your CG model against experimental data, after satisfactory bonded parameters have been achieved, is considered good practice. An in-depth tutorial of parametrizing a new molecule can

$$V^{bonded} = \underbrace{\frac{1}{2}\sum_{bonds} K_r(r-r_0)^2}_{V^{bond}} + \underbrace{\frac{1}{2}\sum_{angles} K_\theta(\theta-\theta_0)^2}_{V^{angle}} + \underbrace{\frac{1}{2}\sum_{n,dihedrals} K_\varphi[1+cos(n\varphi-\delta)]}_{V^{dihedral}}$$

Equation 2 Examples of simple bonded potentials used in Martini: two-body harmonic potential (V^{bond}), three-body angular potential (V^{angle}), and four-body dihedral angle potential (V^{dihedral})

be found under “Tutorials” at the Martini web page (cgmartini.nl), which is summarized in Fig. 1c. Automatic parametrization approaches could be an easier but probably a less accurate option to generate Martini models [32]. However, they could provide prospects for high-throughput simulation methodologies [33]. Note that, to keep the secondary structure of proteins and nucleotides close to the target state (e.g., crystal structure), Martini makes uses of additional harmonic bonds that define an elastic network (*see* **Note 5**).

In the next subsection of this chapter, we will describe a hands-on tutorial, which uses one of our current projects (lipoplex-membrane fusion) as a guide. This practice addresses useful examples of how to build Martini CG models of macromolecules (double-stranded DNA), (solvated) liquid crystals, and complex asymmetric membranes. At the final part, all these CG models are put together for particle-membrane fusion simulations.

2 Hands-On: Cationic Lipid-DNA Lipoplexes for Gene Transfer

This section will be a guide for setting up a CG simulation of a lipoplex-membrane fusion experiment making use of the Martini 2.0 force field with the DNA extension [11]. We will start with a small introduction regarding lipoplexes and their biological relevance. However, the main objective of this section is to show the construction of such a complex system using the Martini approach, from A to Z.

Lipoplexes are complexes of genetic material and lipids used for transfection in gene therapy. Due to the high negative charge of nucleotide polymers (DNA, RNA), they do not readily cross the hydrophobic core of biological membranes via a passive mechanism. Another downside of using naked DNA for therapeutic purposes is its low half-life time in the bloodstream [34]. In lipoplexes, the packing architecture is such that, upon fusion with a membrane or vesicle, its content is mainly released on the side of the membrane opposing the leaflet of initial fusion [35]. This allows the escape of the genetic material from the endosome in which the lipoplex is trapped after being taken up by the cell via endocytosis. Transfection of cells utilizing depathogenized viral vectors currently has a much higher transfection efficacy than the lipid-based vectors or any other non-viral-based method [36]. However, depathogenized viral vectors still trigger the immune system in humans, causing their application in medical gene therapy to be limited [37]. Another drawback of viral vectors compared to their nonviral counterparts is their high preparation cost [37]. Therefore, increasing the efficacy of nonviral vectors for transfection is required for the development of gene therapy as a safe and affordable medical treatment.

In this hands-on section, we will build a lipoplex using one of the earlier lipoplex compositions used [38], in which the DNA is entrapped in an inverted hexagonal (H_{II}) lipid phase. First, the lipoplex itself will be built using 1,2-dioleoyl-sn-glycerol-phosphoethanolamine (DOPE) as the helper lipid and 1,2-dioleoyl-3-trimethylammoniopropane (DOTAP) as the cationic lipid in a 4:1 ratio. In this example, we will use double-stranded (ds) DNA oligomers, with a length of 24 base pairs (bp), as the gene carrier. Replacing the DNA with RNA should not change the general procedure described in this section, though this was not specifically tested. Lastly, the constructed lipolex will be solvated and fused with an asymmetric bilayer mimicking the endosomal membrane.

During this section, lines preceded with a "$" are to be executed in the terminal. This hands-on was designed to be used in combination with GROMACS 2016.3 [39]. Visual Molecular Dynamics (VMD) version 1.9.3 [40] was used for visual inspection, and Python 2.7 was the default python compiler used for all python scripts, unless stated otherwise.

2.1 Building a Liquid Lipoplex Crystal

For building the inverted hexagonal phase (H_{II}), we will make use of the procedures described in refs. [41, 42]. We start with preparing a Martini CG model for DNA, which is to be rested on top of a bilayer. Then, we duplicate this DNA-membrane stack and perform energy minimization, equilibration, and finally a production run to form the crystal lattice.

2.1.1 Preparing the DNA

First, download the .pdb file from the martini website. This structure corresponds to a sequence of DNA with 24 bases in each strand ([CGCGAATTCGCG]$_2$).

```
$ wget http://cgmartini.nl/images/parameters/dna/24bp_AA2CG_
stiff.tar.gz
$ tar -xvf 24bp_AA2CG_stiff.tar.gz
```

To transform an atomistic DNA structure into a CG structure, we will use martinize-dna (cgmartini.nl, under Downloads, Force field parameters, DNA) [11, 12]. Before running martinize-dna, we should remove the ion and water molecules from the .pdb. Be aware that if you have used a DNA structure builder, to obtain a different length or sequence of DNA (e.g., using scfbio-iitd.res.in), the final atom and residue names have to follow the default pdb nomenclature.

The 24bp_cleaned.gro file is already made compatible with the martinize-dna script and we can perform the all-atom (AA) to CG transformation (Fig. 2a). The output includes a CG.pdb and the CG.itp files. We will use a stiff elastic network to imply a strict helical structure to our DNA (*see* **Note 5**). Later on, this could be replaced by a softer elastic network if desired.

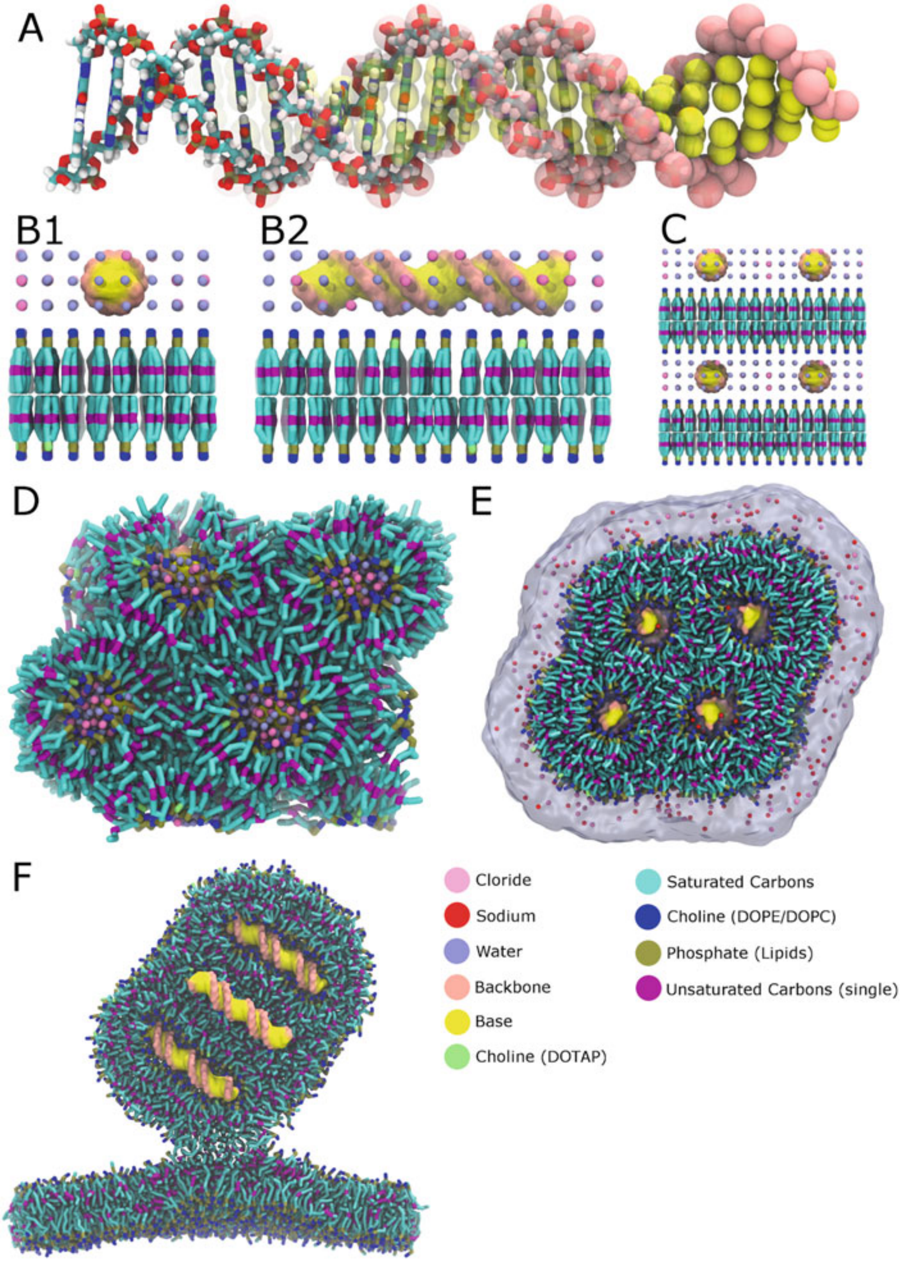

Fig. 2 Building a lipoplex with Martini: (**a**) The CG mapping of a 24 bp dsDNA; (**b1**, **b2**) shows two views of the initial bilayer with a double-stranded DNA; (**c**) This system is replicated to form a lamellar configuration; (**d**) After running 45 ns of MD simulations, the lamellar phase is converted to an inverted H_{II} phase (liquid lipoplex crystal); (**e**) The liquid lipoplex crystal is solvated and coated with lipids; (**f**) An example of a lipoplex-membrane fusion experiment

```
$ python martinize-dna.py -dnatype ds-stiff -f 24bp_cleaned.
gro -x 24bp_CG.pdb
```

Always read the output and check, for example, if the number of chains or base pairs specified matches that what you expect. In our case, this should be two chains (A and B), each containing 24 nucleotides. We need to rotate the DNA such that it will lie parallel to the membrane. We will use GROMACS to do so.

```
$ gmx editconf -f 24bp_CG.pdb -rotate 0 90 0 -o 24bp_CG_ro
tated.gro
```

2.1.2 Creating the DNA Bilayer Stack

Now, we need to generate a simulation box containing our DNA CG model and a symmetric bilayer with the desired lipid composition. We will aim at a 4:1 ratio of DOPE and DOTAP, respectively. To do so, we will use insane [43]. Insane is a python program, developed in-house, which generates an initial CG configuration using a grid-based approach. This procedure makes insane one of the fastest initial state builders for membranes with or without incorporated protein(s). The latest stable version of insane can be downloaded from our web page (cgmartini.nl, under downloads/tools). We will generate a small piece of membrane with DNA, which will be used later on to generate the H_{II} phase with dsDNA inside its channels.

```
$ python insane.py -l DOPE:4 -l DOTAP:1 -alname DOTAP -alhead
'C' -allink 'G G' -altail "CDCC CDCC" -x 11 -y 6.5 -z 7 -f
24bp_CG_rotated.gro -dm 3.5 -o bilayer_1DNA.gro -sol W -salt
0.150 -sold 0.87
```

As DOTAP is not a default lipid, its topology needs to be completely described in insane using the flags "-alname" (name of the new lipid), "-alhead" (groups in the lipid head, where "C" defines the head as choline), "-allink" (groups in the lipid linkers, with "G G" defining two ester groups), and "-altail" (define the lipid tails, where each "C" indicates four carbon atoms in a linear saturated chain, while "D" indicates four linear carbon atoms containing a single unsaturation). This should generate a box containing roughly 187 DOPE, 44 DOTAP, plus the DNA at physiological salt concentration (as shown in Fig. 2b1, b2). The target ratio of atomistic water with respect to the amount of lipids is around 8:1 [41, 42]. For Martini water, with one bead representing four water molecules, the target ratio is thus 2:1. Do remember that the charge of the DNA is not yet neutralized and this will be –46. Therefore, overshooting your target amount of water by 46 is recommended, for we will transform those waters into sodium beads. To generate a bigger complex, we will copy the

DNA-membrane configuration along its axis perpendicular to the channel normal (as shown in Fig. 2c).

```
$ gmx genconf -f bilayer_1DNA.gro -o 2bilayer_4DNA.gro -nbox
1 2 2
```

2.1.3 Preparing the Topology Files

Next, we need to generate a top file which matches the "2bilayer_4DNA.gro" composition and order and makes use of the correct topology files. To achieve a charge neutral system, we will reduce the number of waters in the topology by 46 and add 46 sodium ions right underneath (this will transform 46 water molecules into sodium ions). The final topology should resemble the example topology below.

```
#include "martini-dna-150909/martini_v2.1-dna.itp"
#include "martini-dna-150909/martini_v2.0_ions.itp"
#include "martini_v2.0_DOTAP.itp"
#include "martini_v2.0_DOPE_02.itp"

#define RUBBER_BANDS
#include "Nucleic_A+Nucleic_B.itp"
[ system ]
; name
Martini system containing 4 dsDNA and 2 bilayers
[ molecules ]
; name         number
Nucleic_A+Nucleic_B    1
DOPE             89
DOTAP                  22
DOPE             89
DOTAP                  22
W                106
NA               49
CL               47
... another 3 times for the 2bilayer4_DNA.gro
```

2.1.4 Running EM/EQ and Production

Now, we have all the components together, and we can start running an energy minimization, equilibration, and production run. We will not go into much details, but the default settings can be found at cgmartini.nl (under Downloads, Force field parameters, Input parameters). To perform energy minimization, we need to set the integrator to "steep" and 1000 steps should suffice.

```
$ gmx grompp -f em.mdp -c 2bilayer_4DNA.gro -p topol.top -o em.
tpr -maxwarn 1
$ gmx mdrun -deffnm em -v
```

The "-maxwarn" flag is to allow overwriting name mismatches between the .gro and .top file, caused by the DOTAP and added sodium ions. After running the minimization, we need to create an index file (index.ndx) to use for our temperature coupling scheme. We will create a group for all lipids, the DNA, and the water plus ions. For the sake of this tutorial, those groups are called "Lipids," "DNA" and "W_IONS" respectively. The index groups can be easily generated from the em.gro, using "gmx make_ndx."

```
$ gmx make_ndx -f em.gro -o index.ndx
```

Before we run the equilibration, we need to fix the DNA in the *x* dimension. This will help with a smooth conversion from the periodic crystal to the solvated naked lipoplex. To do so, we need to add a few lines of code to the bottom of our "Nucleic_A +Nucleic_B.itp."

```
#ifdef CONSTRAINED_X
[ position_restraints ]
; ai funct fcx    fcy    fcz
1    1    500     0      0 ; restrains to a plane (y-z-plane)
305  1    500     0      0 ; restrains to a plane (y-z-plane)
#endif
```

For equilibration, 250,000 steps at a 2 fs time step should suffice. We will use anisotropic pressure coupling, and the berendsen barostat for improved stability. Do not forget to set the integrator back to "md" and add "define = -DCONSTRAINED_X". In the "mdrun" command we add an "rdd" of 2 to prevent instabilities due to our long elastic bonds in the DNA.

```
$ gmx grompp -f eq.mdp -c em.gro -p topol.top -n index.ndx -o
eq.tpr
$ gmx mdrun -deffnm eq -v -rdd 2.0
```

For the production run, we will use roughly the same settings as those for equilibration. Important is that the pressure coupling will be changed from "berendsen" to "parrinello-rahman," and the time step for the integrator should be larger. For systems containing DNA, a time step of 10 fs is the maximum (*see* **Note 6**).

```
$ gmx grompp -f md.mdp -c eq.gro -p topol.top -n index.ndx -o
md.tpr
$ gmx mdrun -deffnm md -v -rdd 2.0 -nsteps -1
```

Keep running the simulation until inspection with VMD shows full formation of the H_{II} phase.

While writing this hands-on section, it took roughly 2,268,000 steps (~45 ns CG time, which should correspond to ~200 ns real time, as explained in **Note 7**) to complete the formation of the inverted H_{II} phase (as shown in Fig. 2d). The final frame can be extracted using VMD, and for the sake of this tutorial, we assume it was named md.gro. We are now finished with the construction of the liquid lipoplex crystal. Do keep in mind that this crystal was made to be solvated, and therefore, its PBC conditions were not optimized for a perfect hexagonal unit cell.

2.2 Solvating a Liquid Lipoplex Crystal

For the second part of this tutorial, we will use the liquid lipoplex crystal we generated in part 2.1. The goal is to extract the crystal in such a manner that the channel geometry is not disturbed (naked lipoplex). We will also have to add an extra layer of lipids around our extracted lipoplex to act as a coat (coated lipoplex, as shown in Fig. 2e). Once we have added the coating lipids, we can solvate the whole system. From there on, we energy minimize, equilibrate, and run a production run in a similar manner as that described in part 2.1.

2.2.1 Extracting the Periodic Crystal

The liquid lipoplex crystal generated before contains four channels with four strands of dsDNA in total. This could be considered the smallest possible crystal under cubic periodic boundary conditions. To generate a larger crystal, we use "gmx genconf," to copy this box in the desired dimensions. Even though we will stick to the 2 × 2 geometry for the solvated lipoplex, we will duplicate the box in its "*y*" and "*z*" dimensions. By doing so, we can make extraction of a 2 × 2 lipoplex much easier. This might not seem logical as of yet, but bear with us, and you will see that this is indeed the case. To prevent weird indexes after copying the box, we will first make the .gro file "whole."

```
$ gmx trjconv -f 2bilayer_4DNA.gro -pbc whole -o 2bilayer_4D
NA_whole.gro
$ gmx genconf -f 2bilayer_4DNA_whole.gro -o 4bilayer_16DNA.gro
-nbox 1 2 2
```

We will use VMD to select the inner four strands of DNA. Then, we will make an area selection around these strands of DNA to include all their lipids, water, and ions.

```
$ vmd 4bilayer_16DNA.gro
     "same resid as within 20 of index
     your_central_4_DNA_strands"
```

Export the selected structure and name it "naked_lipoplex.gro."

2.2.2 Coating the Naked Lipoplex

After extracting the naked lipoplex, we will have to add coating lipids. This can be achieved in many ways. One of the options is to increase the selection range in the VMD command used above in combination with excluding any DNA beads. However, the lipids you will have selected in doing so are in a structure that is rather stable. This would make equilibration of the coat of the lipoplex, potentially, a long and expensive process. Therefore, we will make use of PACKMOL, another tool for initial state building [44]. PACKMOL differs from insane, for it uses a packing optimization in defined regions of space. This makes PACKMOL much slower than insane for building simple membranes, but it allows for the addition of molecules to an already complex geometry. PACKMOL can be downloaded free of charge at the web page of the University of Campinas (www.ime.unicamp.br/~martinez/packmol/).

We need to coat the lipoplex with a nice monolayer, and we will make a rough initial approximation by regarding each face of the naked lipoplex as an independent plane. Then we will use the average area per lipid (APL) to calculate the amount of lipids we would need to cover the total area of our lipoplex. To calculate the APL of our mixture, we have to set up a symmetrical bilayer containing our lipid concentrations. A bilayer of 10 × 10 nm^2 will be large enough to get an accurate APL for the lipids used in this tutorial. This bilayer can be constructed following the same protocol as that described in part 2.1. To calculate the APL, we need to simulate the solvated bilayer (under semi-isotropic pressure coupling) up to the point that the box dimensions are stable for a while. Then we can extract the box dimensions over time using “gmx energy”.

```
$ gmx energy -f md.edr
```

Select either the x or y dimension as the preferred output and calculate the average value over the period where it is stable (e.g., the last 10%).

$$\text{APL}_{\text{dope,dotap}} = \frac{2x^2}{\text{Number of lipids}} \approx 0.65\ \text{nm}^2$$

To roughly calculate the area the outer monolayer (the coat) has to cover, you can use VMD. Pressing “2” in the visualization screen will allow you to select two particles and measure the distance between them. We will add 2 nm to each side to accommodate for the fact that the outer leaflet has an increased distance to span with respect to the inner leaflet (which we are measuring).

$$\text{Lipids}_{\text{added}} = \frac{2([(x+2)(y+2)] + [(x+2)(z+2)] + [(y+2)(z+2)])}{\text{APL}_{\text{dope,dotap}}}$$

$$\approx 1800 \Rightarrow 1440\ \text{DOPE}, 360\ \text{DOTAP}$$

After estimating the amount of lipids we need, we will use PACKMOL to configure them around our lipoplex. Extensive tutorials for using PACKMOL can be found at their web page (m3g.iqm.unicamp.br/packmol/). As PACKMOL makes use of .pdb files, we will have to convert our individual files to .pdb.

```
$ gmx editconf -f your_file.gro -o your_file.pdb
```

Use VMD to obtain a .pdb file for each of the lipids using the export coordinates function, with a single lipid selected. After generating the correct PACKMOL input file we can run it.

```
$ packmol < coating_lipoplex.inp
```

Running PACKMOL can take up quite some time and sometimes no correct packing can be found, even upon running for multiple hours. Try playing around with the excluded and included volume which are defined in the input file (outside and inside box), until a satisfactory packing has been achieved. The output structure is presumed to be named "coated_lipoplex.pdb." Convert the .pdb to a .gro in the same manner as was demonstrated in part 1.

2.2.3 Tidying Up the GRO and Topology Files

To tidy things up, you need to sort the "coated_lipoplex.gro" and make sure that the topology has the same order and number of molecules as your sorted .gro. After sorting and updating our topology, we will solvate the lipoplex using insane.

```
$ python insane.py -f coated_lipoplex.pdb -o
solvated_lipoplex.gro -sol W -salt 0.15 -pbc cubic
```

Add the amount of waters and ions insane added to your sorted topology. During the PACKMOL step, we also added more DOTAP, which causes our system to have a nonzero net charge. Therefore, we will replace some of the added water in the insane step with CL to obtain a net charge of zero. To prevent freezing of water (*see* **Note 1**), we will also replace 10% of the waters in our last entry with antifreeze water (WF).

2.2.4 Running EM/EQ and Production

To finish, we need to energy minimize and equilibrate our system before we can start the production run. To do so, we can use the same procedure as before, but we will now use the "-DCONSTRAINED_XYZ" flag (constructed in the same manner as before, but now in XYZ) until large deformations are resolved, and we will use isotropic pressure coupling. It will take roughly 10 μs for the solvated lipoplex to equilibrate its outer coat. You can use "gmx gyrate" to inspect if the lipoplex' shape has stabilized. If the outer coat is too loose, or tight, try adding more or less coating lipids with PACKMOL, or change the included and excluded volume. We are now done with solvating a liquid lipoplex crystal.

2.3 Lipoplex Fusion with an Asymmetric Complex Membrane

For the lipoplex fusion with an asymmetric complex bilayer, we will use the solvated and equilibrated lipoplex generated in part 2.2 of this hands-on section, in combination with a generated asymmetric bilayer.

2.3.1 Creating an Asymmetric Bilayer

By now, we are able to generate complex bilayers; however, we never attempted to generate an asymmetric one. Here, we want to model an endosomal type membrane containing a 1:1 DPPC/DOPC mixture in the upper leaflet and a 2:2:1 DPPC/DOPC/DOPS mixture in the bottom one. To do so, we will use insane again. However, first we generate each leaflet of our asymmetric bilayer as a symmetrical bilayer. Thus, for each asymmetric bilayer, you need to run two symmetric simulations. From those simulations, we can obtain the complex APL for each of the leaflets. To do so, we can use the same protocol as described in part 2.1. After obtaining the complex APL for each of the leaflets, we will use insane to generate the complex bilayer using the obtained APL of each leaflet. This will make sure that the final tension in the two leaflets is equal. An example is given below:

```
$ python insane.py -u DPPC:1 -u DOPC:1 -ua APL$_{DPPC-DOPC}$ -l DPPC:2
-l DOPC:2 -l DOPS:1 -a APL$_{DPPC-DOPC-DOPS}$ -x 10 -y 10 -z 10 -sol W
-salt 0.150 -o complex_asymmetrical_bilayer.gro
```

The topology files for a wide range of lipids can be found at our web page (cgmartini.nl, under Downloads). Make the corresponding topology using the output of insane and equilibrate the membrane. This will take roughly 500 ns, though the amount of time equilibration will take increases with the complexity of the membrane. Equilibration of the membrane can be done as before in parts 2.1 and 2.2; do make sure that you use semi-isotropic pressure coupling and that your membrane lies in the *xy* plane.

2.3.2 Combining the Asymmetric Bilayer and the Lipoplex

After obtaining an equilibrated membrane, we will remove the waters and ions. We now have an equilibrated membrane in vacuum and a solvated lipoplex. To be able to combine the lipoplex and the membrane, we will also extract the solvated lipoplex from its environment. To do so, we will use VMD.

```
$ vmd solvated_lipoplex_equilibrated.gro
```

To extract the lipoplex from its environment, we will use the following selection and export it, as described in part 2.1.

"pbwithin 15 of (resname DOPE DOTAP or name "SC.∗" "BB.∗")" We will save it as "extracted_lipoplex.gro".

To combine our extracted lipoplex with the equilibrated asymmetric complex membrane, we create a combined .gro file. To do so, we copy the content of the membrane into the new .gro file ("lipoplex-membrane.gro"). Remove the final line (the box description). Add the content of the "extracted_lipoplex.gro" but remove the header line, atom count, and box description.

If the lipoplex and membrane overlap, you can use "gmx editconf -translate x y z" to shift your lipoplex. Solvate the system as before using insane. Generate a topology file that matches the . gro one. Energy minimize and equilibrate (as before in part 1) using semi-isotropic pressure coupling. The equilibrated system should resemble Fig. 2f. To perform an unbiased fusion experiment, you have to simulate for a long time to observe adhesion, stalk formation, and transfection. However, such a simulation would take up a large amount of computational time; therefore, you could add biasing forces to drive membrane adhesion and initiate stalk formation. From there on, you could remove all biasing forces to observe the fusion process after initial stalk formation [1].

2.3.3 Alternative Methods and Tools

We suggest to the reader as possible follow-up steps for this hands-on section, the building of even more complex CG systems. For example, a lipoplex fusion experiment within a vesicle (a process that could mimic the DNA transfection event in the early endosome). However, the current implementation of insane does not allow us to build such lipid structures yet. A user-friendly option for this problem could be the usage of a graphical user interface as CHARMM-GUI Martini Maker [45, 46]. In the current implementation, this program can build Martini models of micelles, nanodiscs, bilayers, and vesicles. We hope that the tools and tips presented above will help you on your way with your own implementations of the Martini forcefield. If you would like to learn more about Martini, you can visit our web page for other tutorials (cgmartini.nl). For any questions regarding the implementation of the Martini force field for your project(s), we would like to direct you to the forum at our web page.

3 Outlook

Since its initial publications, the Martini force field has been developed and tested in a broad range of applications, from simple lipid bilayers to complex fusion processes as detailed here for the lipoplex hands-on. Despite the huge success of the model, certain problems have been reported as excessive protein and sugar aggregation (*see*

Note 4), and water freezing (also described in **Note 1**). Along with the modeling demand of new and challenging systems, these limitations pushed the MD group of Groningen to improve the CG beads—the fundamental building blocks of Martini—until now largely untouched since version 2. One of the main features of the forthcoming version, entitled Martini 3, is the re-parametrization of small (S) and tiny (T) beads, designed to be fully balanced with the normal (N) size beads. New chemical-type beads were also tuned to model systems not included in the current version. For instance, we will have more beads with hydrogen-bonding capabilities (including all polar and nonpolar beads) and charged beads dedicated for modeling divalent ions. Water has also its own special bead, parametrized to improve its miscibility with other beads and also avoid freezing problems. The interaction matrix was modified, including more interaction levels and smoother transitions between the beads. In this aspect, special attention was taken regarding charged beads, with trends in solvent polarization and ion-pi interactions implicitly included in the Lennard-Jones potential with neutral beads. Together with other new features of the model, preliminary tests indicate a significant improvement of proteins, sugars, and nucleic acids in Martini 3. Besides, exciting new systems seem to nicely behave, including MD simulations of ionic liquids, and coacervates, as well as protein-ligand binding. We conclude that a promising new era of Martini CG simulations is coming.

4 Notes

This last section of the chapter contains a series of notes, which include useful information, limitations, and tips for problems that can arise using the Martini force field.

1. *Limited stability of fluid phase and water freezing problem:* This is a known consequence of the use of the LJ 12–6 potential for the nonbonded interactions. The thermodynamic behavior of solid-fluid and gas-fluid interfaces should therefore be interpreted with care, at least at the quantitative level. In applications where such interfaces are formed (especially water-vapor), these limitations have to be kept in mind. In biomolecular simulations, a related problem is the potential freezing of the Martini water model. The LJ parameters for water (5.0 kJ/mol and 0.47 nm) put the model into the solid-state region of the LJ phase diagram. However, the use of a shifted and truncated potential reduces the long-range attractive part and the CG water is more fluid compared to the standard LJ particle. While the freezing temperature is higher than it should be (around 290 K, [3, 5, 47]), in most applications, freezing is not observed as long as no nucleation site is formed. At lower

temperatures, rapid freezing is a potential problem in systems where a nucleation site is already present (such as an ordered bilayer surface) or when periodicity enhances the long-range ordering. In these cases, a simple pragmatic solution is the addition of 10% antifreeze particles into the bulk water [5].

2. *Entropy-enthalpy compensation:* Martini parameterization is based on partition free energies. The inherent entropy loss on coarse graining is necessarily compensated for by a reduced enthalpy term [24]. The enthalpy/entropy balance of many processes may therefore be biased when modeled at the CG level and affect its temperature dependence, although not necessarily weakening it. For instance, the temperature-dependent hydration free energy for linear alkanes was found to be more pronounced in the CG representation compared to an AA representation [24]. As is true for any force field, applications outside the temperature range used for parameterization (270–330 K) have to be considered with care. Although absolute entropies are clearly underestimated due to the loss of atomistic degrees of freedom, entropy differences can still be accurate [48].

3. *"Sticky problem" in sugars and proteins:* In the past years, the Martini force field has showed some specific limitations involving excessive interactions between certain classes of molecules. For example, protein-protein interactions in water solution seem to be overestimated [49]. To a lesser extent, this effect was also demonstrated for some transmembrane proteins [50]. Recently, mono, oligo- and polysaccharides were found to aggregate in simulations even at moderate concentrations, below their solubility limit [51]. These similar problems (called together here as "sticky problems") could be attenuate for pragmatic solutions, as down-scaling of the Lennard-Jones parameters between the solutes [49–51]. Another option is to increase the interactions with the solvent, which was successfully applied to study protein-crowded environments [52]. The usage of S-beads for the modeling of carbohydrate rings seems to reduce their aggregation propensity, as shown in the re-parametrization of gangliosides [53]. All these possible procedures are not ideal solutions, as they are applied without a deeper understanding of the reasons behind the sticky problem. Besides, scaling factors are specific for classes of molecules and could potentially change important properties of the systems, as the correct binding mode of proteins dimers [50]. New rules for the usage of S- and T-beads together with re-parametrization of Martini (e.g., including specific cross-interactions between standard and S/T bead sizes) showed to be crucial to reduce the excessive interactions (results not published yet). In the near future, all these features will be released as new version of the Martini force field (coined as version 3.0).

4. *Electrostatic interactions and implicit screening*: Another difficulty encountered in our CG model, and perhaps in most coarse-graining approaches, is the correct modeling of the partitioning of polar and charged compounds into a low dielectric medium. Because of the implicit screening, the interaction strength of polar substances is underestimated in nonpolarizable solvents. Applications involving the formation of polar/charged complexes in a nonpolar environment are especially prone to be affected. The inability to form a transmembrane water pore upon dragging a lipid across the membrane is an example [5, 54]. The development of a Martini water model that includes orientational polarization by the means of a dipole represented by two drude charges attached to each water bead allows to correct for some of these effects [30]. Apart from the implicit screening in the CG model, the neglect of long-range electrostatic forces poses a further limitation. Pairwise interactions beyond 1.1 nm (between 2 and 3 CG beads away) are not taken into account. In principle, long-range electrostatic interactions could be added to the CG model, in ways similar to those used in atomistic simulations [31]. In particular, PME in combination with the polarizable Martini water model is often used.

5. *Fixed structure for proteins and nucleic acids:* In applications of peptides, proteins, and nucleic acids, one has to be aware that structure transformations are not modeled in the current parameterization. For proteins, the secondary structure (SS) is essentially fixed by the use of bond angle and dihedral angle potential energy functions. The backbone bead type is also a function of the SS, to take into account the fact that when involved in interactions stabilizing a given element, the backbone is less prompted to engage in other interactions. The backbone interaction strength is therefore decreased when involved in a SS element. This approach allows discrimination between various secondary structure elements but prevents realistic transitions between them. Processes in which folding and unfolding are playing a substantial role are therefore not suitable for modeling with the current Martini force field. However, movements of SS elements with respect to each other are possible and were shown to be quite realistic, for instance, in modeling the gating of a membrane-embedded mechanosensitive channel [55]. In cases where the specificity of the local deformations of the protein backbone is important, the use of other approaches are necessary, as combining Martini with an elastic network [56, 57] or with structure-based CG models [58]. In the case of peripheral membrane proteins, further corrections in the side-chain dihedral angles could also be necessary [59]. Martini DNA and RNA can be used

to model both single and double-stranded structures. For the single-stranded settings, the structure is considered flexible, while double-stranded could be modeled with two different elastic networks: a soft model which has a cutoff of 1.2 nm and a force constant of 13 kJ/mol/nm^2, and the stiff model which has a cutoff of 1.0 nm and a 500 kJ/mol/nm^2 force constant [11, 12].

6. *Time step*: Martini has been parameterized using time steps in the range of 10–40 fs. Whether you can use 40 fs or have to settle for a somewhat smaller time step depends on your system, and on your attitude toward coarse-grained modeling, as explained below. First, the Martini force field is not an atomistically detailed force field. Many assumptions underlie the model, the major one being the neglect of some of the atomistic degrees of freedom. As a result, the interactions between particles are effective ones and the energy landscape is highly simplified. This simplified energy landscape allows for a greatly increased sampling speed at the cost of a loss of detail. This makes CG models in general so powerful. The emphasis, therefore, should not be to sample the energy landscape as accurately as possible, but rather, as effectively as possible. This is in contrast to traditional all-atom models, for which the energy landscape is more realistic and an accurate integration scheme is more important. In practice, the inherent "fuzziness" of the Martini model makes the presence of small energy sinks or sources a less critical problem than in accurate atomistic simulations. Second and most importantly, structural properties are rather robust with respect to the time step; for a time step up to 40 fs, there are no noticeable effects on structural properties of the systems investigated. Moreover, thermodynamic properties such as the free energy of solvation also appear insensitive to the size of the time step. Thus, if the goal is to generate representative ensembles quickly, large time steps seem acceptable. Whereas one can debate the first argument (i.e., the "idealist" vs. "pragmatic" view of the power of CG simulations), the second argument (i.e., the insensitivity of both structural and thermodynamic properties to the magnitude of the time step) implies that a reduction of the time step to 10 fs or below, as has been suggested [60], is a waste of computer time [47]. Nevertheless, time steps of 40 fs and beyond may be pushing the limits too far for certain systems. For some systems, as nucleic acids, time steps higher than 10 fs promote simulation instability [11, 12]. We therefore recommend a time step of 10–20 fs to be on the safe side. Of course, one should always check whether or not results are biased by the choices made. Given that the largest simplifications are made at the level of the interaction potentials, this can best be done by comparing to results obtained using more detailed models.

7. *Effective timescale:* The CG dynamics are faster than the AA dynamics, because the CG interactions are much smoother compared to atomistic interactions. The effective friction caused by the fine-grained degrees of freedom is missing. Based on comparison of diffusion constants for a range of systems (including simple solvents and lipids) in the CG model versus experimental data, the effective time sampled using CG interactions is three- to eightfold larger. When interpreting the simulation results with the CG model, a standard conversion factor of 4 has been used, which is the effective speed-up factor in the diffusion dynamics of CG water compared to real water. The same order of acceleration of the overall dynamics is also observed for a number of other processes, including the permeation rate of water across a membrane [3], the sampling of the local configurational space of a lipid [61], the aggregation rate of lipids into bilayers [3], and the self-diffusion of lipids [3, 5], transmembrane peptides [62], and proteins [63]. However, the speed-up factor can be quite different in other systems or for other processes, and in general no simple conversion of the time axis can be performed. Particularly for protein and nucleic acid systems, no extensive testing of the actual speed-up due to the CG dynamics has been performed, although protein translational and rotational diffusion was found to be in good agreement with experimental data in simulations of CG rhodopsin [63]. In general, the timescale of the simulations has to be interpreted with care.

Acknowledgments

The authors would like to thank the many people who have directly and indirectly contributed to the development of the Martini force field. In particular Alex de Vries, Helgi I. Ingolfsson, Manuel N. Melo, Tsjerk Wassenaar, Xavier Periole and all the past and present members of the MD group in Groningen, as well as the many users abroad, are acknowledged for their dynamism and enthusiasm in using, criticizing, and improving Martini.

References

1. Marrink SJ, Mark AE (2003) The mechanism of vesicle fusion as revealed by molecular dynamics simulations. J Am Chem Soc 125:11144–11145
2. Marrink SJ, Mark AE (2003) Molecular dynamics simulation of the formation, structure, and dynamics of small phospholipid vesicles. J Am Chem Soc 125:15233–15242
3. Marrink SJ, de Vries AH, Mark AE (2004) Coarse grained model for semiquantitative lipid simulations. J Phys Chem B 108:750–760
4. Marrink SJ, Risselada J, Mark AE (2005) Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model. Chem Phys Lipids 135:223–244
5. Marrink SJ, Jelger Risselada H, Yefimov S et al (2007) The MARTINI force field: coarse

grained model for biomolecular simulations. J Phys Chem B 111:7812–7824

6. Risselada HJ, Marrink SJ (2008) The molecular face of lipid rafts in model membranes. Proc Natl Acad Sci U S A 105:17367–17372
7. Monticelli L, Kandasamy SK, Periole X et al (2008) The MARTINI coarse-grained force field: extension to proteins. J Chem Theory Comput 4:819–834
8. de Jong DH, Singh G, Bennett WFD et al (2013) Improved parameters for the Martini coarse-grained protein force field. J Chem Theory Comput 9:687–697
9. López CA, Rzepiela AJ, de Vries AH et al (2009) Martini coarse-grained force field: extension to carbohydrates. J Chem Theory Comput 5:3195–3210
10. López CA, Bellesia G, Redondo A et al (2015) MARTINI coarse-grained model for crystalline cellulose microfibers. J Phys Chem B 119:465–473
11. Uusitalo JJ, Ingólfsson HI, Akhshi P et al (2015) Martini coarse-grained force field: extension to DNA. J Chem Theory Comput 11:3932–3945
12. Uusitalo JJ, Ingólfsson HI, Marrink SJ et al (2017) Martini coarse-grained force field: extension to RNA. Biophys J 113:246–256
13. de Jong DH, Liguori N, van den Berg T et al (2015) Atomistic and coarse grain topologies for the cofactors associated with the photosystem II core complex. J Phys Chem B 119:7791–7803
14. Panizon E, Bochicchio D, Monticelli L et al (2015) MARTINI coarse-grained models of polyethylene and polypropylene. J Phys Chem B 119:8209–8216
15. Rossi G, Fuchs PFJ, Barnoud J et al (2012) A coarse-grained MARTINI model of polyethylene glycol and of polyoxyethylene alkyl ether surfactants. J Phys Chem B 116:14353–14362
16. Rossi G, Monticelli L, Puisto SR et al (2011) Coarse-graining polymers with the MARTINI force-field: polystyrene as a benchmark case. Soft Matter 7:698–708
17. Alessandri R, Uusitalo JJ, de Vries AH et al (2017) Bulk heterojunction morphologies with atomistic resolution from coarse-grain solvent evaporation simulations. J Am Chem Soc 139:3697–3705
18. Liu J, Qiu L, Alessandri R et al (2018) Enhancing molecular n-type doping of donor-acceptor copolymers by tailoring side chains. Adv Mater 30(7):1704630
19. Qiu L, Liu J, Alessandri R et al (2017) Enhancing doping efficiency by improving host-dopant miscibility for fullerene-based n-type thermoelectrics. J Mater Chem A Mater Energy Sustain 5:21234–21241
20. Grunewald F, Rossi G, de Vries AH et al (2018) Transferable MARTINI model of poly (ethylene oxide). J Phys Chem B 122:7436–7449
21. Monticelli L (2012) On atomistic and coarse-grained models for C60 fullerene. J Chem Theory Comput 8:1370–1378
22. Ingolfsson HI, Tieleman P, Marrink S (2015) Lipid organization of the plasma membrane. Biophys J 108:358a
23. Van Eerden FJ, Melo MN, Frederix PWJM et al (2017) Exchange pathways of plastoquinone and plastoquinol in the photosystem II complex. Nat Commun 8:15214
24. Baron R, Trzesniak D, de Vries AH et al (2007) Comparison of thermodynamic properties of coarse-grained and atomic-level simulation models. ChemPhysChem 8:452–461
25. Shih AY, Arkhipov A, Freddolino PL et al (2006) Coarse grained protein–lipid model with application to lipoprotein particles. J Phys Chem B 110:3674–3684
26. Shih AY, Freddolino PL, Arkhipov A et al (2007) Assembly of lipoprotein particles revealed by coarse-grained molecular dynamics simulations. J Struct Biol 157:579–592
27. Periole X, Marrink S-J (2013) The Martini coarse-grained force field. Methods Mol Biol 924:533–565
28. de Jong DH, Baoukina S, Ingólfsson HI et al (2016) Martini straight: boosting performance using a shorter cutoff and GPUs. Comput Phys Commun 199:1–7
29. López CA, de Vries AH, Marrink SJ (2013) Computational microscopy of cyclodextrin mediated cholesterol extraction from lipid model membranes. Sci Rep 3:2071
30. Yesylevskyy SO, Schäfer LV, Sengupta D et al (2010) Polarizable water model for the coarse-grained MARTINI force field. PLoS Comput Biol 6:e1000810
31. Michalowsky J, Schäfer LV, Holm C et al (2017) A refined polarizable water model for the coarse-grained MARTINI force field with long-range electrostatic interactions. J Chem Phys 146:054501
32. Bereau T, Kremer K (2015) Automated parametrization of the coarse-grained Martini force field for small organic molecules. J Chem Theory Comput 11:2783–2791
33. Menichetti R, Kanekal KH, Kremer K et al (2017) In silico screening of drug-membrane thermodynamics reveals linear relations between bulk partitioning and the potential of mean force. J Chem Phys 147:125101

34. Kawabata K, Takakura Y, Hashida M (1995) The fate of plasmid DNA after intravenous injection in mice: involvement of scavenger receptors in its hepatic uptake. Pharm Res 12:825–830

35. Zhao Y, Huang L (2014) Lipid nanoparticles for gene delivery. Adv Genet 88:13–36

36. Ramamoorth M (2015) Non viral vectors in gene therapy - an overview. J Clin Diagn Res 9:Ge01–Ge06

37. Chira S, Jackson CS, Oprea I et al (2015) Progresses towards safe and efficient gene therapy vectors. Oncotarget 6:30675–30703

38. Ciani L, Ristori S, Salvati A et al (2004) DOTAP/DOPE and DC-Chol/DOPE lipoplexes for gene delivery: zeta potential measurements and electron spin resonance spectra. Biochim Biophys Acta 1664:70–79

39. Abraham MJ, Murtola T, Schulz R et al (2015) GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1–2:19–25

40. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14:33–38, 27–28

41. Marrink S-J, Mark AE (2004) Molecular view of hexagonal phase formation in phospholipid membranes. Biophys J 87:3894–3900

42. Corsi J, Hawtin RW, Ces O et al (2010) DNA lipoplexes: formation of the inverse hexagonal phase observed by coarse-grained molecular dynamics simulation. Langmuir 26:12119–12125

43. Wassenaar TA, Ingólfsson HI, Böckmann RA et al (2015) Computational lipidomics with insane: a versatile tool for generating custom membranes for molecular simulations. J Chem Theory Comput 11:2144–2155

44. Martínez L, Andrade R, Birgin EG et al (2009) PACKMOL: a package for building initial configurations for molecular dynamics simulations. J Comput Chem 30:2157–2164

45. Qi Y, Cheng X, Im W (2015) CHARMM-GUI Martini maker for coarse-grained simulations. Biophys J 108:161a

46. Hsu P-C, Bruininks BMH, Jefferies D et al (2017) CHARMM-GUI Martini Maker for modeling and simulation of complex bacterial membranes with lipopolysaccharides. J Comput Chem 38:2354–2363

47. Marrink SJ, Periole X, Peter Tieleman D et al (2010) Comment on "On using a too large integration time step in molecular dynamics simulations of coarse-grained molecular models" by M. Winger, D. Trzesniak, R. Baron and W. F. van Gunsteren, Phys. Chem. Chem. Phys., 2009, 11, 1934. Phys Chem Chem Phys 12:2254

48. Yano Y, Matsuzaki K (2006) Measurement of thermodynamic parameters for hydrophobic mismatch 1: self-association of a transmembrane helix. Biochemistry 45:3370–3378

49. Stark AC, Andrews CT, Elcock AH (2013) Toward optimized potential functions for protein-protein interactions in aqueous solutions: osmotic second virial coefficient calculations using the MARTINI coarse-grained force field. J Chem Theory Comput 9:4176–4185

50. Javanainen M, Martinez-Seara H, Vattulainen I (2017) Excessive aggregation of membrane proteins in the Martini model. PLoS One 12: e0187936

51. Schmalhorst PS, Deluweit F, Scherrers R et al (2017) Overcoming the limitations of the MARTINI force field in simulations of polysaccharides. J Chem Theory Comput 13:5039–5053

52. Liu B, Åberg C, van Eerden FJ et al (2017) Design and properties of genetically encoded probes for sensing macromolecular crowding. Biophys J 112:1929–1939

53. Gu R-X, Ingólfsson HI, de Vries AH et al (2017) Ganglioside-lipid and ganglioside-protein interactions revealed by coarse-grained and atomistic molecular dynamics simulations. J Phys Chem B 121:3262–3275

54. Bennett WFD, Peter Tieleman D (2011) Water defect and pore formation in atomistic and coarse-grained lipid membranes: pushing the limits of coarse graining. J Chem Theory Comput 7:2981–2988

55. Melo MN, Arnarez C, Sikkema H et al (2017) High-throughput simulations reveal membrane-mediated effects of alcohols on MscL gating. J Am Chem Soc 139:2664–2671

56. Periole X, Cavalli M, Marrink S-J et al (2009) Combining an elastic network with a coarse-grained molecular force field: structure, dynamics, and intermolecular recognition. J Chem Theory Comput 5:2531–2543

57. Siuda I, Thøgersen L (2013) Conformational flexibility of the leucine binding protein examined by protein domain coarse-grained molecular dynamics. J Mol Model 19:4931–4945

58. Poma AB, Cieplak M, Theodorakis PE (2017) Combining the MARTINI and structure-based coarse-grained approaches for the molecular dynamics studies of conformational transitions in proteins. J Chem Theory Comput 13:1366–1374

59. Herzog FA, Braun L, Schoen I et al (2016) Improved side chain dynamics in MARTINI

simulations of protein–lipid interfaces. J Chem Theory Comput 12:2446–2458

60. Winger M, Trzesniak D, Baron R et al (2009) On using a too large integration time step in molecular dynamics simulations of coarse-grained molecular models. Phys Chem Chem Phys 11(12):1934–1941

61. Baron R, de Vries AH, Hünenberger PH et al (2006) Configurational entropies of lipids in pure and mixed bilayers from atomic-level and coarse-grained molecular dynamics simulations. J Phys Chem B 110:15602–15614

62. Ramadurai S, Holt A, Schäfer LV et al (2010) Influence of hydrophobic mismatch and amino acid composition on the lateral diffusion of transmembrane peptides. Biophys J 99:1447–1454

63. Periole X, Huber T, Marrink S-J et al (2007) G protein-coupled receptors self-assemble in dynamics simulations of model bilayers. J Am Chem Soc 129:10126–10132

Methods in
Molecular Biology 2022

Springer Protocols

Massimiliano Bonomi
Carlo Camilloni *Editors*

Biomolecular Simulations

Methods and Protocols

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, UK

For further volumes:
http://www.springer.com/series/7651

Biomolecular Simulations

Methods and Protocols

Edited by

Massimiliano Bonomi

Structural Bioinformatics Unit, Institut Pasteur, CNRS UMR 3528, Paris, France

Carlo Camilloni

Dipartimento di Bioscienze, Università degli Studi di Milano, Milano, Italy

Humana Press

Editors
Massimiliano Bonomi
Structural Bioinformatics Unit
Institut Pasteur, CNRS UMR 3528
Paris, France

Carlo Camilloni
Dipartimento di Bioscienze
Università degli Studi di Milano
Milano, Italy

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-4939-9607-0 ISBN 978-1-4939-9608-7 (eBook)
https://doi.org/10.1007/978-1-4939-9608-7

This Humana imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.
The registered company address is: 233 Spring Street, New York, NY 10013, U.S.A.