

Methods in
Molecular Biology 2199

Springer Protocols

Yu Wai Chen
Chin-Pang Benu Yiu *Editors*

Structural Genomics

General Applications

Second Edition



 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>

Structural Genomics

General Applications

Second Edition

Edited by

Yu Wai Chen

Department of Applied Biology and Chemical Technology and the State Key Laboratory of Chemical Biology and Drug Discovery, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Chin-Pang Benu Yiu

BMI Biotechnology Appraisals & Investments Limited, Wan Chai, Hong Kong

 **Humana Press**

Editors

Yu Wai Chen
Department of Applied Biology
and Chemical Technology and
the State Key Laboratory of Chemical
Biology and Drug Discovery
The Hong Kong Polytechnic University
Hung Hom, Hong Kong

Chin-Pang Bennu Yiu
BMI Biotechnology Appraisals & Investments Limited
Wan Chai, Hong Kong

ISSN 1064-3745

ISSN 1940-6029 (electronic)

Methods in Molecular Biology

ISBN 978-1-0716-0891-3

ISBN 978-1-0716-0892-0 (eBook)

<https://doi.org/10.1007/978-1-0716-0892-0>

© Springer Science+Business Media, LLC, part of Springer Nature 2014, 2021

Chapter 11 is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapter.

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Humana imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.

The registered company address is: 1 New York Plaza, New York, NY 10004, U.S.A.



Protocol for Simulations of PEGylated Proteins with Martini 3

Fabian Grünewald, Peter C. Kroon, Paulo C. T. Souza, and Siewert J. Marrink

Abstract

Enhancement of proteins by PEGylation is an active area of research. However, the interactions between polymer and protein are far from fully understood. To gain a better insight into these interactions or even make predictions, molecular dynamics (MD) simulations can be applied to study specific protein-polymer systems at molecular level detail. Here we present instructions on how to simulate PEGylated proteins using the latest iteration of the Martini coarse-grained (CG) force-field. CG MD simulations offer near atomistic information and at the same time allow to study complex biological systems over longer time and length scales than fully atomistic-level simulations.

Key words Martini, Proteins, Polymers, PEGylation, Coarse-grained simulation, Modified proteins

1 Introduction

Since polyethylene glycol (PEG) was for the first time covalently attached to proteins in the late 1970s, this strategy, known as PEGylation, has become a viable tool for enhancing proteins [1]. For example, PEGylation is known to increase the stability of the secondary structure against temperature [2], increase circulation time of protein therapeutics [3], and decrease immune responses [3]. At the same time, it usually does not interfere with protein activity or secondary structure [1–6]. However, it has also been found that the PEG conformation is intimately connected to the efficiency of mentioned enhancements [7, 8]. Generally, it is assumed that the PEG chain adopts one of two possible conformational motifs: the so-called “shroud” conformation or the “dumbbell” conformation. In the shroud conformation, PEG wraps around the protein. In the dumbbell conformation, however, PEG exists as a coil next to the protein resembling one end of the dumbbell with the other end being the protein [4]. Overall, there

appears to be no consensus as to whether one conformation is favored over the other. Both direct and indirect evidence exists for either hypothesis and it seems to depend on molecular weight, the specific protein, as well as how many PEG chains are attached [4]. For example, in their hallmark study, Pai and coworkers used small-angle neutron scattering (SANS) to directly measure the polymer conformation of PEGylated lysozyme and human growth hormone. Their study shows that PEG assumes the dumbbell type conformation, where it exists as a coil next to the protein [4]. In a more recent study, LaCouer and coworkers showed that the activity of PEGylated hemoglobin peaks with a PEG chain length of 10 kg/mol. They hypothesized, also based on SANS experiments, that this change in activity is caused by PEG transitioning from the dumbbell to the shroud conformation, where the polymer wraps around the protein. However, it was not possible to find a sharp crossover point as function of increasing chain length where this conformational change would occur [8].

Molecular dynamics (MD) simulations have also been used to investigate the conformations of PEG in PEGylated proteins [9]. MD simulations are a powerful tool because they offer insight into biomolecular assemblies at the molecular level. Recently, Munasinghe and coworkers studied PEGylated Albinum Bovin Serum (BSA) using atomistic MD simulations and observed a conformational change with increasing chain length. They state that this is likely to be driven by strong interactions with specific amino acids. These interactions only become relevant as the PEG length increases, thus driving the conformational change [7]. If this proves true, it would mean that the conformations of PEGs and thus their enhancement capabilities are protein specific. However, as pointed out by Lin, Ramezanghorbani, Colina and coworkers, atomic level detailed simulations are limited in timescale, length, and complexity [7, 9]. To overcome these limitations and potentially have high throughput screenings, coarse-grained (CG) molecular dynamics simulations can be used.

CG simulations with the Martini force-field [10] are among the most popular for biomolecular applications. They have been widely applied to study complex biological systems such as the plasma membrane [11] and the thylakoid membrane including the light-harvesting complex II [12]. For a practical view on the Martini force-field, see the relevant chapter in the same book series [13]. Here we only recount the basic details of the model and its latest iteration (i.e., Martini 3).

The Martini force-field utilizes a building-block approach. Chemical moieties or small molecules of 2–5 non-hydrogen atoms are represented as one particle, called bead. Beads are the minimal building blocks and can be combined together to represent larger molecules. Each bead has a type, which defines how it interacts with the other beads in the force-field and its size. The

type is chosen from a predefined set of types. The best type is selected by closely matching experimental free energies of transfer from water to organic solvents of the underlying chemical fragment [10]. However, with Martini 3 other experimental properties such as miscibility are increasingly used for the selection and validation as well [14]. Bonded interactions between the beads, such as bond distances, angles and dihedral angles, are optimized to best represent the underlying molecular geometry, volume, and flexibility. They are derived by reproducing reference probability distributions obtained from atomistic simulations [15].

Using this approach, CG models for many biological [16–18] and synthetic molecules [19–22] have been created. One of the strengths of the Martini force-field is the compatibility between these different models. For example, it is no problem to combine PEG with proteins to represent PEGylated proteins as has been done before with Martini [23–25]. However, as detailed recently, Martini 2 has some pitfalls and drawbacks [26]. For instance, no standard bead was able to represent PEG with sufficient accuracy. This has led various authors to create special beads for PEG, which in turn limited their compatibility when used under different circumstances than the authors had designed it for [21]. A different drawback of Martini 2 are the overestimated interactions of proteins with each other [26] and potentially incorrect PEG protein interactions [23]. To overcome these limitations, we recommend the new Martini 3 force-field, which was specifically designed to increase the compatibility and to represent a wider variety of chemical fragments accurately. It has already been shown to overcome some of the drawbacks of the original Martini model [14, 15, 21, 27, 28].

In the next section, we will work through and explain how to generate parameters and input structures for PEGylated proteins using Martini 3. However, this guide can also be taken as an example for generating parameters and structures for simple proteins (Subheading 2.1) and polymers (Subheading 2.2). In Subheading 3, a detailed protocol for setting up a simulation and equilibrating it will be presented. Finally, Subheading 4 comprises useful practical tips and information.

2 Martini Parameters for PEGylated Proteins

Usually PEGylated proteins consist of (1) a protein, which is unmodified with respect to its native state [1] (i.e., preserved secondary structure and amino acid sequence), (2) a linker fragment attached to an amino acid, and (3) the PEG polymer chain. Following the Martini building-block approach, parameters for each of the parts are generated separately and then combined. In the last step an input structure for the PEGylated protein is

generated. Subheading 3 subsequently shows how to setup a simulation and equilibrate it. The flow chart in Fig. 1 shows this process in more detail. Whenever appropriate, we will discuss choices of parameters, input file formats and program options in a more general manner. In this way, the reader can adopt the options to their target problem.

2.1 Software Requirements

All the necessary input files to follow this tutorial can be downloaded from our website (<http://www.cgmartini.nl/>) or our Git-Hub page (<https://github.com/marrink-lab>). The tutorial requires a working installation of GROMACS (version 2016 or higher) [29], python 3, martinize 2 (<https://github.com/marrink-lab/vermouth-martinize>), and polyply (https://github.com/marrink-lab/polyply_1.0). For visualization, any program which can visualize gro files and PDB files, such as VMD or pymol, can be used. All programs required are open source. As detailed in **Note 1**, the tutorial—with minimal modification—can also be run on Windows or Mac OS. Note that commands, which need to be executed in the terminal are preceded by “\$.” All basic commands can also be found next to the flow chart (Fig. 1) as quick reference.

2.2 Martini Parameters for Simple Proteins

As example protein, we have chosen a mono-PEGylated lysozyme as used in the study of Pai and coworkers [4]. To begin, an appropriate structure (i.e., PDB code LZ3T) of lysozyme needs to be downloaded from the protein data bank. As discussed in **Note 2**, it is important to make sure that it is complete and contains all non-hydrogen atoms. Once the structure is obtained, the program martinize 2 [30] will be used to generate both CG itp files and starting structures.

2.2.1 Martinizing Lysozyme

Download the PDB file using the following command:

```
$ wget http://www.rcsb.org/pdb/files/3lzt.pdb
```

Martinize 2 requires definitions of the force-field and mappings of the amino acids. Files with these definitions are shipped with martinize 2, so only the name of the force-field (i.e., “-ff martini30b32”) needs to be provided. Furthermore, the atomistic PDB file of the protein, downloaded in the previous step, is required. The following command generates the basic parameters and a coarse-grained structure file.

```
$ martinize2 -f 3lzt.pdb -ff martini30b32 -x lysozyme_cg.pdb -o topol.top
```

In general, the basic options above should be supplemented by few more to generate appropriate parameters for Martini 3 proteins:

Example Commands

```

$ wget www.rcsb.org/pdb/files/3lzt.pdb

$ martinize2
-f 3lzt.pdb
-ff martini30b32
-x lysozyme_cg.pdb
-o topol.top
-dssp -scfix -cys auto
-elastic -p backbone

$ gmx editconf
-f lysozyme_cg.pdb
-o lysozyme_cg.gro
-box 8.5 8.5 8.5

$ polyply gen_itp
-f PEO.martini.3b.itp
  OH_end.itp OH_link.ff
-seq PEO:50 OHend:1
-o PEG_50_OH.itp
-name PEGOH

$ polyply gen_itp
-f molecule_0.itp MEE.itp
  PEG_50_OH.itp methoxy_link.ff
-seq molecule_0:1 MEE:1 PEGOH:1
-o lysoPEG.itp
-name lysoPEG

$ polyply gen_coords
-p system.top
-o lysoPEG.gro
-name lysoPEG
-c lysozyme_cg.gro

$ gmx solvate
-cp lysoPEG.gro
-o solvated.gro
-cs water.gro
-radius 0.21

$ gmx grompp
-f min.mdp
-c solvated.gro
-p system.top
-o dummy.tpr

$ gmx genion
-f solvated.gro
-o start.gro
-s dummy.tpr
-conc 0.15
-neutral

$ EM_EQ_run.sh
    
```

Flowchart

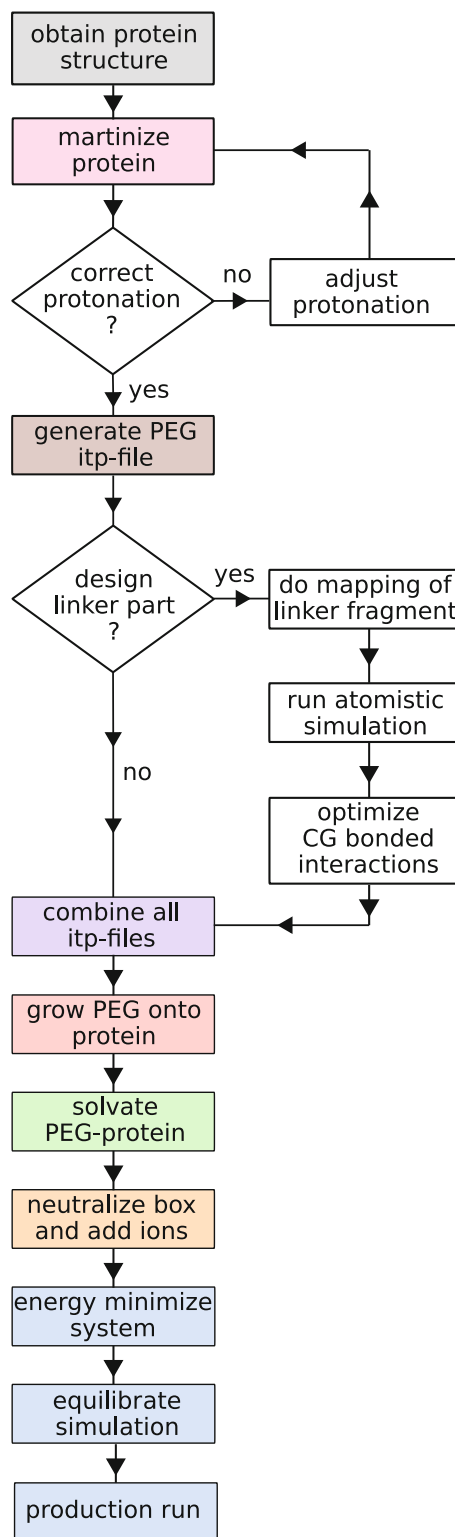


Fig. 1 Process of generating parameters and input structures for PEGylated proteins with Martini. The left column shows all commands required for the example

“*-dssp*” invokes the program DSSP [31, 32] to analyze the secondary structure of the reference structure in PDB file. This gives the necessary input to generate the bonded parameters of the protein model, which are secondary structure dependent [16]. It is not always appropriate to define the secondary structure based on a reference structure. In this case, you can define it manually as explained in **Note 3**.

“*-elastic*” is the option used to generate an elastic network to maintain the tertiary structure. In general, there are two different options for maintaining the tertiary structure: the elastic network approach [33], and the Go approach [34]. For this tutorial, the elastic network approach will be used. As outlined in **Note 4**, the Go approach can be used for more accurate protein dynamics.

“*-scfix*” tells martinize to apply the side chain fix. Herzog and coworkers have shown that including additional dihedral angle potentials for the amino acid side chains improves the protein dynamics [35]. The dihedrals are defined based on the reference atomistic structure and applied to the whole protein. So, in case you are simulating an unfolded peptide, you should not use the *-scfix* option. This so-called side chain fix (ScFix) is used by default for Martini3 folded proteins and applied for this tutorial.

“*-cys*” is used to let martinize 2 determine whether cysteine bridges exist. If the program finds a cysteine bridge, it will include the interactions that link the two beads participating in the cysteine bridge. Whereas linking PEG via the thiol group of cysteine is a very popular method for PEGylating proteins, the ligation usually targets cysteines that are not involved in cysteine bridges [1]. Therefore, this option can safely be used for all PEGylated proteins unless PEGylation is specifically known to disrupt a cysteine bridge. In that case, the relevant interactions need to be removed from the itp file manually after running martinize 2.

“*-p backbone*” can be used to generate position restraints for the protein backbone beads. These restraints are useful for the initial equilibration step. For example, it allows the water to penetrate cavities, which otherwise would collapse quickly. Because later, when this equilibration procedure is applied, these initial parameters are required.

Combining all the options above will generate a standard Martini 3 lysozyme protein with elastic network as well as a coarse-grained structure file. The final command is the following:

```
$ martinize2 -f 3lzt.pdb -ff martini30b32 -x lysozyme_cg.pdb
-dssp -elastic -scfix -cys auto -p backbone -o topol.top
```

2.2.2 Checking Protonation States

Regular Martini 3 uses fixed protonation states, which means titratable amino acids are either neutral or charged. The pH of the simulation is usually assumed to be at physiological pH (i.e., pH 7.4). However, it should always be verified that the protonation states of the titratable amino acids are correct. The easiest way is to inspect the CG itp file. Aspartic acid, Glutamic acid, Lysine, Histidine, Tyrosine, Cysteine, the C-terminus, and the N-terminus can in principle change their protonation state. Their protonation state together with the number of Arginines (which are always charged at pH 7.4) determines the total charge of the protein, which can also serve as an indication for the protonation states. The following two commands will print all titratable amino acids and compute the total charge.

```
$ egrep 'ASP | LYS | GLU | HIS | TYR | CYS' molecule_0.itp
$ grep Q molecule_0.itp | awk '{sum += $7} END {print sum}'
```

Inspecting the output will show that lysozyme has a total charge of +8 and that all titratable amino acids except Tyrosine are charged. This is consistent with titration experiments of lysozyme [36, 37]. It is well known that pK_a values of amino acids can change as a result of their local environment. Therefore, in the absence of experimental data, the pK_a values of the amino acids in the protein should at least be estimated (e.g., using the H++ server <http://biophysics.cs.vt.edu/>) or more advanced tools and models should be used (*see* **Note 5**). To modify the protonation state with martinize 2, first a PDB file with the accurate protonation states needs to be obtained. For example, such a file can be downloaded from one of the servers performing the estimates. Make sure the format adheres to the specifications outlined in **Note 2**. Subsequently, the residue names of the amino acids, with changed protonation state, have to be changed according to the following scheme: Add a zero in front of the name and delete the last letter. For example, the residue name of aspartic acid changes from “ASP” to “0AS.” Using the modified PDB file, martinize can be run again to get the parameters for the protein with accurate protonation states.

2.3 Martini Parameters for PEG

In this section, we will show how to generate input parameters for Martini PEG using the program polyply. Polyply can be used to generate structures and input parameters for linear polymers or stitch together any existing itp files.

2.3.1 Homopolymer

To generate itp files for any polymer, polyply needs a monomer itp file (GROMACS format). This monomer itp file needs to contain all atoms part of the monomer repeat unit within the “[atom-types]” directive. Furthermore, the file needs to include all the

bonded interactions that all the atoms listed before have with all following atoms. The idea is outlined in Fig. 2. Consider situation A, where we want a polymer of four monomers. Our input file should contain atom 1 in the “[atomtypes]” directive and all the interactions of 1 with the next monomers (indicated by arrows). The idea of this format is that the parameters for the rest of the monomers can be obtained by shifting the initial monomer (blue) one to the right and generating all bonded interactions accordingly. So the interaction between 1 and 2 becomes an interaction between 2 and 3 (Fig. 2b). This would generate an interaction between 2 and 5. However, since our chain only has four monomers this interaction would be dropped. As an example, the monomer input file for the Martini3 beta-version of PEG [21] is shown below. The PEG repeat unit is $-\text{[CH}_2\text{-O-CH}_2\text{]}-$ and modeled as one bead.

```
[ moleculetype ]
; name nexcl.
PEO 1
[ atoms ]
1 SN1a 1 PEO EO 1 0.000 45
[ bonds ]
1 2 1 0.37 7000
[ angles ]
1 2 3 2 135.00 50
1 2 3 10 135.00 75
[ dihedrals ]
1 2 3 4 1 180.00 1.96 1
1 2 3 4 1 0 0.18 2
1 2 3 4 1 0 0.33 3
1 2 3 4 1 0 0.12 4
```

Because Martini PEG is one bead per repeat unit, the “[atomtypes]” directive only has one atom. Furthermore, it has one bond with the next monomer, two angles, and four dihedral angle terms involving the next three monomers. Hence the highest atom index, which needs to be included in this itp file, is 4. Note that all monomer itp files always need to start with atom index 1. Polyply is also provided with a number of default monomer itp files (*see Note 6*). To generate a PEG of 3 kDa length (e.g., ~50 repeat units) only the monomer itp file, the sequence of monomers and the name of the molecule have to be provided as well as a name for the newly generated itp file. The sequence is provided using the “-seq” flag and consists of one or more blocks of the format “residue name: number of monomers.” The residue name must match at most one molecule name in the itp files provided with the “-f” flag.

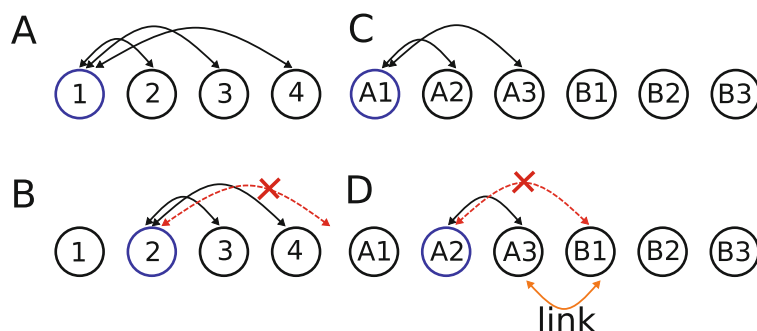


Fig. 2 Schematic of itp file generation using polyply; arrows indicate interactions (e.g., bonds, angles, etc.) between monomer fragments indicated by circles. Itp files of the polymer are generated from a single monomer itp file. This itp file (blue circle panel **a**) needs to define all interactions of that fragment with the next monomers (1–4 panel **a**). Then these interactions are used to generate the new interactions (panel **b**), but all interactions exceeding the maximum number of atoms (red arrow) are not taken into account. Similarly, in the case of block-copolymers the A block is built from a single monomer of each block (A1 and B1 panel **c**). When the interactions are generated, all interactions exceeding the single block (red arrow panel **d**) are not taken into account. This makes it necessary to supply an extra file defining all links between blocks (orange arrow, panel **d**)

```
$ polyply gen_itp -f PEO.martini.3b.itp -seq PEO:50 -o PEG_50.itp -name PEG
```

2.3.2 End Group and Block-Copolymers

Often, the end of a homopolymer chain is different from the monomeric repeat unit. For PEG, for example, one can either have a $\text{CH}_2\text{-CH}_3$ or $\text{CH}_2\text{-OH}$ group at the end. Especially for small PEG molecules the choice is important because the influence of the end group is proportionally larger than for longer chains. To illustrate the itp file generation for the case when an end group needs to be attached, the PEG chain will be terminated with a $\text{CH}_2\text{-OH}$ end group.

Attaching an end group follows the same procedure as attaching another block to form a block copolymer. All that is needed is another monomer itp file for the block and a link file, which uses the vermouth force-field format (i.e., “.ff”) [30]. This concept is illustrated in Fig. 2c, d. Polyply will generate the bonded interaction from the monomer itp files for each block separately, removing any overlapping interactions between the blocks. For example, the interaction between A1 and A3 would generate a new interaction between A2 and B1. However, since B1 belongs to a different block, the interaction is removed. Therefore, a second itp file specifying all the interactions linking the two blocks has to be supplied. This link file adheres to the vermouth force-field format. For use with polyply, each link file needs to contain the same first three lines. Those lines are shown in the example file below. These lines tell the program that it is a link and that the interactions listed

directly apply to the final polymer. After these three lines are defined all interactions specifying the link with the correct atom numbers as in the final polymer should follow. The link file cannot contain the “[moleculetypes]” and “[atomtypes]” directive. Note that the indices in the bond and angle directive do not start at 1 but are exactly the same as in the final polymer. For instance, for the end group, the link file should look as follows:

```
[ link ]
[ molmeta ]
by_atom_id true
[ bonds ]
50 51 1 0.280 7000.0
[ angles ]
49 50 51 2 140.00 25.0
```

And the CH₂-OH capping group monomer itp file only defines one bead:

```
[ moleculetype ]
; name nexcl.
OHend 1
[ atoms ]
1 TP1 1 OHend EO 1 0.000 36
```

This procedure of stitching together two itp files will also be used to combine the protein itp file and the PEG itp file together in the end. For now, the following command can be used to combine the PEG_50.itp with the end group. Note that the file extension of the link file is “.ff” and not “.itp.”

```
$ polyply gen_itp -f PEG_50.itp OH_end.itp OH_link.ff -seq
PEG:1 OHend:1 -o PEG_50_OH.itp -name PEGOH
```

The same result could also be obtained using one command:

```
$ polyply gen_itp -f PEO.martini.3b.itp OH_end.itp OH_link.ff -seq
PEO:50 OHend:1 -o PEO_50_OH.itp -name PEGOH
```

2.4 Linking PEG to the Protein

So far we have shown how to generate parameter files for lysozyme and PEG. However, before these are combined, it has to be defined how the PEG chain is attached to the protein. Attachment of PEGs to proteins is often done via the amine group of a Lysine or the N-terminus, or the thiol group of a Cystein [1].

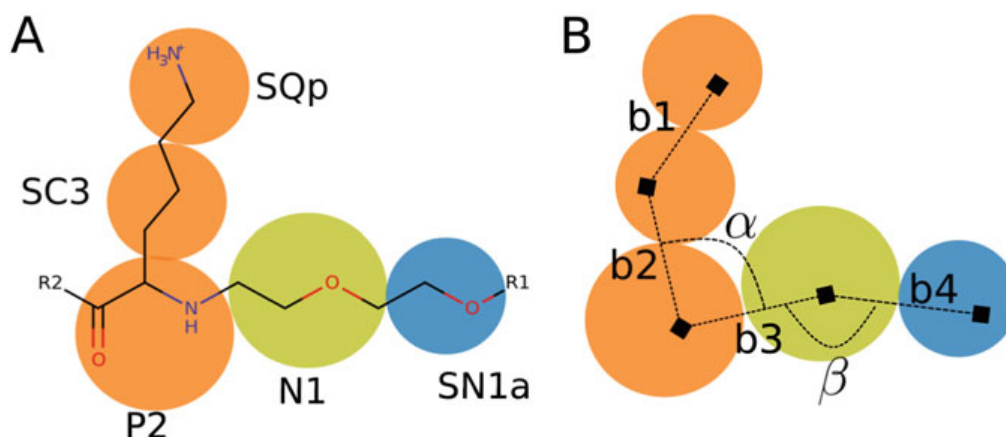


Fig. 3 Methoxyethane (yellow) linking PEG (R1, one repeat unit shown in blue) to the N-terminus of the protein (R2, terminal residue shown in orange). Panel **a** shows the mapping (i.e., each circle is one bead) and bead types. Panel **b** shows the definition of the bonded interactions b1–b4, α , and β

2.4.1 Designing Martini Parameters for the Linker

The linker motif used in this example is shown in Fig. 3a. It is the result of reductive alkylation, which is used to selectively attach PEG to the N-terminus. To design parameters for this linker, a mapping has to be designed which splits the linker into beads. The mapping is shown as circles in Fig. 3a. There is only one bead between the N-terminus and the PEG chain. Note that the mapping of the N-terminal amino acid (i.e., lysine) becomes the same as for the backbone in the rest of the protein. Thus the default backbone bead type (i.e., P2) should be used for this bead. The bead linking PEG and protein corresponds to methoxyethane. Therefore, the bead type that best resembles methoxyethane has to be found. The free energy of transfer from octanol to water of methoxyethane was estimated to be around 3.4 kJ/mol [10]. An N1 bead has a free energy of transfer of 4.1 kJ/mol and is thus the best match. The remaining beads are the regular PEG beads. Bonded interactions (i.e., bonds and angles), were obtained by simulating a fragment (R1=CH₃, R2=CH₂-O-CH₃) in water at the atomistic level using GROMOS [38] parameters obtained from the ATB [39] and reproducing the probability distributions at the CG level. We will not go into more detail for designing the linker, as it follows the normal design rules for Martini (*see Note 7*).

2.4.2 Combining Protein, Linker and PEG Parameters

Next, all itp files are combined together: first, the bead type of the N-terminal Lysine has to be changed to a neutral P2 bead. Open the lysozyme.itp file and change the type of the first bead to P2 and the charge to zero. The beginning of your edited file should look as follows:

```
[ moleculetype ]
molecule_0 1
```

```
[ atoms ]
1 P2 1 LYS BB 1 0.0
2 SC3 1 LYS SC1 2 0.0
3 SQp 1 LYS SC2 3 1.0
```

Next the itp file for the methoxyethane linker is defined. As it is only one bead, it looks the same as for the CH₂-OH end group.

```
[ moleculetype ]
; name nexcl.
MEE 1 ; methoxyethane-link
[ atoms ]
1 N1 1 MEE MEE 1 0.000 72
```

As done for the CH₂-OH end group before, a link file has to be created. The link file has to contain all bonded interactions that span the N-terminus, methoxyethane, and the first PEG bead. Figure 3b shows these interactions. Note that b1 and b2 are already defined in the protein itp file. In contrast to the file above defining the monomeric repeat unit, the link file uses the indices of the final itp file. In this example, the itp files are combined in the order: protein-linker-PEG. Thus the N-terminus will be the first bead with index 1. Because lysozyme has 292 beads, the methoxyethane linker bead will have the atom index 293 and the first PEG bead will have the index 294. Therefore, the link itp file needs to look as follows:

```
[ link ]
[ molmeta ]
by_atom_id true
[ bonds ]
1 293 1 0.41 2000 ; b3
293 294 1 0.39 5000 ; b4
[ angles ]
293 1 2 2 150 15 ; alpha
294 293 1 2 170 50 ; beta
```

Finally, all the files can be combined to obtain an itp file for the PEGylated lysozyme:

```
$ polyply gen_itp -f molecule_0.itp MEE.itp PEG_50_OH.itp methoxy_link.ff -seq
molecule_0:1 MEE:1 PEGOH:1 -o lysoPEG.itp -name lysoPEG
```

If all the files had already been generated, the links for the OH end group and methoxyethane could have been combined into one

file. Then it would have been possible to generate the itp file using a single command:

```
$ polyply gen_itp -f molecule_0.itp MEE.itp PEO.martini.3b.itp OH_end.itp
  combined_links.ff -seq molecule_0:1 MEE:1 PEG:50 OHend:1 -o lysoPEG.itp
  -name lysoPEG
```

However, it is common to use multiple invocations rather than doing everything at once.

2.5 Generation of Input Structures

Having obtained an itp file for PEGylated lysozyme, a starting structure can be generated. The protein structure generated by martinize 2 is supplied to polyply, which will add a PEG chain to it. Polyply only reads gro files. Thus the martinize 2 PDB file needs to be converted to gro format. In addition, the box size for the system can already be specified. As further detailed in **Note 8**, a sufficiently large box size should be used in order to stay below the overlap concentration. A cube with sides of 8.5 nm is sufficient for this purpose.

```
$ gmx editconf -f lysozyme_cg.pdb -o lysozyme_cg.gro -box 8.5 8.5 8.5
```

Polyply also requires an accurate topology file including the same information as used to run the simulation in vacuum. The topology file for PEGylated lysozyme will look as follows:

```
#include martini3/martini_v3.0.4.itp
#include lysoPEG.itp
[ system ]
lysoPEG in water
[ molecules ]
lysoPEG 1
```

This topology file has to be provided to polyply together with the name of the polymer and some other options as shown below:

```
$ polyply gen_coords -p system.top -o lysoPEG.gro -name lysoPEG
  -c lysozyme_cg.gro
```

Using the approach shown above polyply can in principle generate a starting conformation for any CG polymer with a few limitations as outlined in **Note 9**. Because polyply cannot generate the protein structure, the one obtained from martinize 2 is reused. Using this command an input structure as shown in Fig. 4 is generated. To generate just a PEG chain in vacuum, it would have been possible to omit the “-c” option and not define any initial structure to reuse.

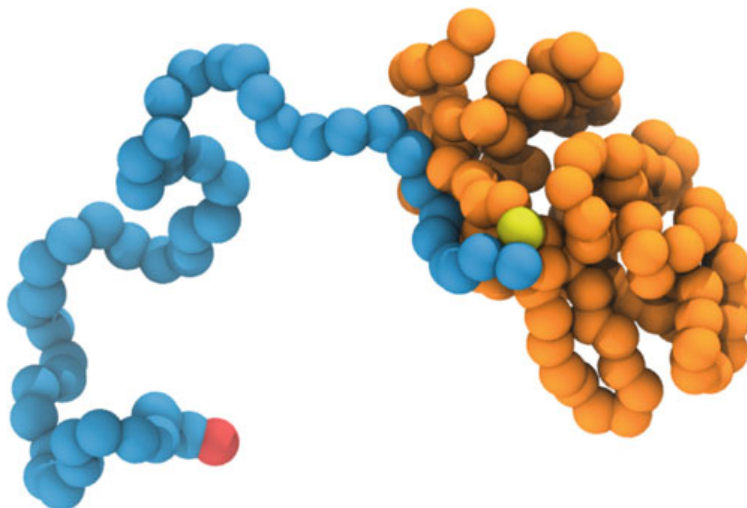


Fig. 4 PEGylated lysozyme initial structure; PEG is shown in blue, the linker bead in yellow and the protein backbone in orange

3 Setting Up and Running a Simulation

Once the starting structure for the PEGylated protein is generated, the system can be set up. Before solvating, the system should be energy minimized applying both positional restraints to the protein backbone and the polymer. This allows the initial structure to relax, but prevents it from coiling up. The positional restraints and the flexible options can be switched on by including the following line in the mdp file. All mdp files are provided with the tutorial files. General mdp files appropriate for Martini can be downloaded from our website (<http://www.cgmartini.nl/>).

```
define = -DFLEXIBLE -DPOSRES
```

Now run:

```
$ gmx grompp -f min.mdp -p system.top -c lysoPEG.gro -o min.tpr -r lysoPEG.gro
$ gmx mdrun -v -deffnm min
```

Next, the energy-minimized structure is solvated with water using the GROMACS tool `gmx solvate`:

```
$ gmx solvate -cs water.gro -cp min.gro -o protein_water.gro
  -radius 0.21 2>&1 | tee solv.out
```

This will add water into the simulation box. It is important to set the “-radius 0.21” option to account for the fact that Martini water is representing four water molecules at a time. If it is not set, GROMACS will pack too many water beads, which might lead to

instabilities during the equilibration. Because solvate adds water molecules to our system, the topology file needs to be revised. The following two commands will do it automatically.

```
$ water=$(grep "W (" solv.out | awk '{print $5}')
$ echo "WN ${water}" >> system.top
```

Besides water, salt needs to be added to the simulation box for two reasons: (1) As the protonation states are fixed, the solution needs to be neutralized; (2) to better mimic the biological environment of proteins, simulations of soluble proteins are generally run at 150 mM salt concentration. Of course, this can be adjusted to reproduce other experimental concentrations. The program `gmX genion` needs a `tpr` file to generate the ions and modify the topology file. For this purpose, a dummy `tpr` file using the energy minimization settings as before can be composed. The following commands generate the final box.

```
$ gmX grompp -f min.mdp -p system.top -c protein_water.gro
-o dummy.tpr -r protein_water.gro -maxwarn 1
$ echo WN | gmX genion -s dummy.tpr -neutral -conc 0.15
-p system.top -o start.gro
```

Before starting the production simulation, a series of energy minimization and equilibration should be run. This is especially important for large proteins and polymers. First, the final box is energy minimized using flexible bonds and position restraints.

```
$ gmX grompp -f min.mdp -p system.top -c start.gro -o min.tpr -r start.gro
$ gmX mdrun -v -deffnm min
```

Next, a short equilibration of 50 ns applying positional restraints is run. This equilibration allows the water to solvate the polymer and protein. Furthermore, through the use of the Berendsen barostat [40], the simulation will quickly relax to the final volume.

```
$ gmX grompp -f eq.mdp -p system.top -c min.gro -o eq.tpr
-r min.gro -maxwarn 1
$ gmX mdrun -v -deffnm eq
```

Finally, the positional restraints need to be released and another equilibration simulation using the Berendsen barostat should be run. Here the Berendsen barostat is used, as the simulation is more stable than with Parrinello–Rahman [41] pressure coupling.

```
$ gmX grompp -f eq2.mdp -p system.top -c eq.gro -o eq2.tpr -maxwarn 1
$ gmX mdrun -v -deffnm eq2
```

Now, everything is in place to perform the final production run. At this stage, the Parrinello–Rahman pressure coupling [41] should be used. However, before starting the simulation, a few properties should be checked: First open the output file in VMD [42] or any other visualization software and check whether the protein and polymer conformations look reasonable.

```
$ vmd eq2.gro
```

Next, it is useful to compute the box pressure, and temperature average to confirm that all have reached the intended target values.

```
$ gmx energy -f eq2.edr -o energy_eq.xvg
$ gmx analyze -f energy_eq.xvg
```

The values you obtain should be close to 1 bar for the pressure, 310 K for temperature, and a box volume which is constant. If they have converged, the production run can be started. Long simulation times are required to sufficiently sample the polymer conformational space. In the past, sampling times of around 10–30 μs were used for polymer systems with Martini [19–21, 43]. For this example, the simulation time is set to 2 μs .

```
$ gmx grompp -f NpT.mdp -p system.top -c eq2.gro -o run.tpr -maxwarn 1
$ gmx mdrun -v -deffnm run
```

After about 2 μs of simulation, the PEGylated lysozyme has a conformation as shown in Fig. 5. It is clearly not in the shroud conformation. It exists as an extended chain fitting to the dumbbell conformation. This is the same conformation as found by Pai and coworkers for 30 kDa PEGylated lysozyme [4].

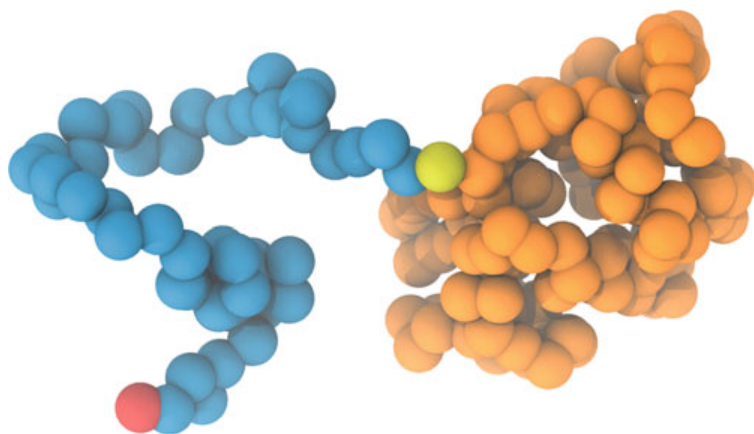


Fig. 5 Conformation of PEGylated lysozyme after 2 μs of simulation

In this tutorial we have generated Martini 3 parameters for PEGylated lysozyme in three steps: (1) martinize 2 was used to obtain parameters for lysozyme; (2) polyply was used to generate parameters for OH terminated PEG-50; (3) polyply was used again to stitch together the protein and polymer parameters as well as the molecular fragment linking the two. It was also shown how to generate a starting structure for this molecule, and set up as well as equilibrate a simulation. Although a specific example was used to explain this procedure, the protocol is general and can be used for any PEGylated protein. More information about Martini and more tutorials are available on our web page (cgmartini.nl). Any questions regarding the realization of your project with martini, can be posed via our forum. We would also like to encourage reporting of bugs and problems regarding martinize 2 and polyply via the GitHub pages of the two programs.

4 Notes

1. To set up and run simulations on Windows or Mac OS, GRO-MACS, and Python 3 have to be installed. Both have appropriate versions available. Because martinize 2 and polyply are python programs, they can be run in operating system independent from within any python 3 environment. Note that in the tutorial, sometimes bash specific commands (e.g., “egrep” or “awk”) are used. These will only work on Linux OS running a bash shell. However, different solutions can easily be found suiting the OS of interest.
2. Martinize 2 poses some requirements on the input PDB file: There can be no missing residues; C and N atoms, which define the peptidic bond, must be present; and atom names should strictly adhere to the PDB format. In addition, residues in the PDB file are identified by their residue names, and the corresponding information must be present in the library of martinize 2. To add nonstandard residues to the library, please refer to a more specialized tutorial. In case of problems, the flags “-write-graph,” “-write-repair,” and “-write-canon” can be used to write out PDB files of the structure as interpreted by the program at various stages.
3. The bonded parameters of Martini proteins are assigned based on the secondary structure. [16] Usually, the secondary structure of a reference crystal structure is used for this purpose. However, this might not always be appropriate. In such cases, the secondary structure can be provided manually to martinize 2 using the “-ss” flag in a text-based format using a single-letter code. Each letter represents the basic secondary structure elements: H = helix, E = sheet/extended, and C = coil/turn.

For example, a tripeptide in a coil-like structure can have its secondary structure defined with the option “-ss CCC.”

4. All Martini protein models need special interactions to maintain the tertiary structure of the proteins. However, different approaches exist to accomplish this goal. The regular elastic network approach simply applies a bond between all backbone beads within a cutoff. This is sufficient to keep the structure stable and has reasonable properties [33]. However, it has been shown that improved flexibility of the protein structure is achieved with the Go [34] or ELNEDIN [33] approaches. Recently with Martini 3, in combination with the Go approach, it has even been shown that an allosteric pathway can be captured [27]. Therefore, if it is suspected that flexibility of certain domains is important for the PEG protein interactions, using a Go approach would improve the simulations.
5. It is common to treat titratable amino acids with fixed protonation state in MD simulations, even if the protonation states can change. However, it is also known that different environments can affect the pK_a of amino acids and thus their protonation state. It could be possible that PEG, especially in the shroud conformation, modifies the protonation states of titratable groups. If this is suspected, you can gain better insight by using a method with dynamic protonation states, which allows amino acids to change in the course of the simulation. We recently have put forward a proof-of-principle for such a method [44] and the GROMACS lambda dynamics approach [45] is also a suitable option.
6. Polyply has a library of default monomer itp files of different coarse-grained and atomistic polymers that come with the program. To see which files are available run “polyply gen_itp -list_lib.” To use these files, you can simply select the name from the list obtained with the previous command and then use the “-lib” flag instead or together with “-f.” For example, to generate the itp file for the PEG-50 polymer, you could also run:

```
polyply -lib martini3_beta -n_mon 50 -o PEG_50.itp -name PEG
```

7. A tutorial for linkers: <http://cgmartini.nl/index.php/tutorials-general-introduction-gmx5/parametrizing-new-molecule-gmx5>
8. When going from the dilute solution regime into the semi-dilute regime, polymer–polymer interactions become important or even dominating. The crossover point is indicated by the overlap concentration. As the concentration increases beyond the overlap concentration also the properties of

polymers change [46]. It is therefore important to choose an appropriate concentration when comparing to experiments. On one hand, if they are conducted in or extrapolated to the dilute solution regime, the simulation box needs to be sufficiently large. On the other hand, as pointed out by Pai et al., the osmotic pressure in cells is often higher than in dilute solution [4]. Therefore, PEGylated proteins should perhaps also be studied under crowded conditions. In this case, it would be appropriate to add more proteins and/or PEG chains to the simulation box to achieve higher concentrations. *See* ref. 21 on how to calculate the overlap fraction for PEG in water.

9. Polyply can generate structures for disordered, arbitrarily complex, polymers. However, this also means that coordinates for any polymer which has a certain long-range order should not be generated this way. Examples include proteins with a secondary structure, DNA, or polymers with large extended aromatic ring fragments. Another practical limitation applies: Polyply cannot generate polymers which contain virtual sides. Examples include Martini 2 P3HT [47].

References

1. Canalle LA, Löwik DWPM, Van Hest JCM (2010) Polypeptide-polymer bioconjugates. *Chem Soc Rev* 39:329–353
2. Pechar M, Kopečková P, Joss L et al (2002) Associative diblock copolymers of poly(ethylene glycol) and coiled-coil peptides. *Macromol Biosci* 2:199–206
3. Milton Harris J, Martin NE, Modi M (2001) Pegylation: a novel process for modifying pharmacokinetics. *Clin Pharmacokinet* 40:539–551
4. Pai SS, Hammouda B, Hong K et al (2011) The conformation of the poly(ethylene glycol) chain in mono-PEGylated lysozyme and mono-PEGylated human growth hormone. *Bioconjug Chem* 22:2317–2323
5. Daly SM, Przybycien TM, Tilton RD (2005) Adsorption of poly(ethylene glycol)-modified lysozyme to silica. *Langmuir* 21:1328–1337
6. Hamley IW (2014) PEG-peptide conjugates. *Biomacromolecules* 15:1543–1559
7. Munasinghe A, Mathavan A, Mathavan A et al (2019) Molecular insight into the protein-polymer interactions in N-terminal PEGylated bovine serum albumin. *J Phys Chem B* 123:5196–5205
8. Le Cœur C, Combet S, Carrot G et al (2015) Conformation of the poly(ethylene glycol) chains in DiPEGylated hemoglobin specifically probed by SANS: correlation with PEG length and in vivo efficiency. *Langmuir* 31:8402–8410
9. Lin P, Colina CM (2019) Molecular simulation of protein-polymer conjugates. *Curr Opin Chem Eng* 23:44–50
10. Marrink SJ, Risselada HJ, Yefimov S et al (2007) The MARTINI force field: coarse grained model for biomolecular simulations. *J Phys Chem B* 111:7812–7824
11. Ingólfsson HI, Melo MN, Van Eerden FJ et al (2014) Lipid organization of the plasma membrane. *J Am Chem Soc* 136:14554–14559
12. Thallmair S, Vainikka PA, Marrink SJ (2019) Lipid fingerprints and cofactor dynamics of light-harvesting complex II in different membranes. *Biophys J* 116:1446–1455
13. Bruininks BMH, Souza PCT, Marrink SJ (2019) A practical view of the Martini force field. In: *Biomolecular simulations, Methods in molecular biology (methods and protocols)*. Springer, New York, pp 105–127
14. Souza PCT et al (2020) Martini 3, submitted
15. Alessandri R (2019) Multiscale modeling of organic materials: from the morphology up. Dissertation, University of Groningen
16. 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

17. Uusitalo JJ, Ingólfsson HI, Akhshi P et al (2015) Martini coarse-grained force field: extension to DNA. *J Chem Theory Comput* 11:3932–3945
18. Lo CA, Rzepiela AJ, De Vries AH et al (2009) Martini coarse-grained force field: extension to carbohydrates. *J Chem Theory Comput* 5:3195–3210
19. 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
20. Panizon E, Bochicchio D, Monticelli L et al (2015) MARTINI coarse-grained models of polyethylene and polypropylene. *J Phys Chem B* 119:8209–8216
21. Grunewald F, Rossi G, de Vries AH et al (2018) Transferable MARTINI model of poly(ethylene oxide). *J Phys Chem B* 122:7436–7449
22. Monticelli L (2012) On atomistic and coarse-grained models for C₆₀ fullerene. *J Chem Theory Comput* 8:1370–1378
23. Ramezanghorbani F, Lin P, and Colina CM (2018) Optimizing protein–polymer interactions in a poly(ethylene glycol) coarse-grained model. *J Phys Chem B* acs.jpcc.8b05359
24. Woo SY, Lee H (2014) Molecular dynamics studies of PEGylated α -helical coiled coils and their self-assembled micelles. *Langmuir* 30:8848–8855
25. Zaghmi A, Mendez-Villuendas E, Greschner AA et al (2019) Mechanisms of activity loss for a multi-PEGylated protein by experiment and simulation. *Mater Today Chem* 12:121–131
26. Alessandri R, Souza PCT, Thallmair S et al (2019) Pitfalls of the Martini model. *J Chem Theory Comput* 15:5448–5460
27. Souza PCT, Thallmair S, Marrink SJ et al (2019) An allosteric pathway in copper, zinc superoxide dismutase unravels the molecular mechanism of the G93A amyotrophic lateral sclerosis-linked mutation. *J Phys Chem Lett* 10:7740–7744
28. Liu J, Qiu L, Alessandri R et al (2018) Enhancing molecular n-type doping of donor–acceptor copolymers by tailoring side chains. *Adv Mater* 30:1–9
29. 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
30. Kroon PC (2020) Automate, aggregate, assemble. Dissertation, University of Groningen
31. Touw WG, Baakman C, Black J et al (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res* 43:D364–D368
32. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22:2577–2637
33. Periolo 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
34. 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
35. 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
36. Kuehner DE, Engmann J, Fergg F et al (1999) Lysozyme net charge and ion binding in concentrated aqueous electrolyte solutions. *J Phys Chem B* 103:1368–1374
37. Bartik K, Redfield C, Dobson CM (1994) Measurement of the individual pK_a values of acidic residues of hen and turkey lysozymes by two-dimensional ¹H NMR. *Biophys J* 66:1180–1184
38. Stroet M, Caron B, Visscher KM et al (2018) Automated topology builder version 3.0: prediction of solvation free enthalpies in water and hexane. *J Chem Theory Comput* 14:5834–5845
39. Malde AK, Zuo L, Breeze M et al (2011) An automated force field topology builder (ATB) and repository: version 1.0. *J Chem Theory Comput* 7:4026–4037
40. Berendsen HJC, Postma JPM, van Gunsteren WF et al (1984) Molecular dynamics with coupling to an external bath. *J Chem Phys* 81:3684–3690
41. Parrinello M, Rahman A (1981) Polymorphic transitions in single crystals: a new molecular dynamics method. *J Appl Phys* 52:7182–7190
42. Humphrey W, Dalke A, Schulten K (1996) VMD: Visual Molecular Dynamics. *J Mol Graph* 14:33–38
43. Rossi G, Barnoud J, Monticelli L (2014) Polystyrene nanoparticles perturb lipid membranes. *J Phys Chem Lett* 5:241–246
44. Grunewald F, Souza PCT, Abdizadeh H, et al (2020) Titratable Martini Model for Constant pH Simulations. *J Chem Phys* 153:024118

45. Donnini S, Tegeler F, Groenhof G et al (2011) Constant pH molecular dynamics in explicit solvent with λ -dynamics. *J Chem Theory Comput* 7:1962–1978
46. Colby RH, Rubinstein M (2003) *Polymer physics*. Oxford University Press, New York
47. 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