

Supporting information for:

PyCGTOOL: Automated Generation of Coarse-grained Molecular Dynamics Models from Atomistic Trajectories

James A. Graham, Jonathan W. Essex, and Syma Khalid*

School of Chemistry, University of Southampton, Hampshire, SO17 1BJ, United Kingdom

E-mail: S.Khalid@soton.ac.uk

Software Dependencies

The software requires Python3 with the following packages available from *pip*:

- Numpy - <http://www.numpy.org/>
- simpletraj - <https://github.com/arose/simpletraj>

The following modules are optional dependencies:

- MDTraj for pseudo-CG XTC output - <http://mdtraj.org/1.7.2/>
 - Scipy - <https://www.scipy.org/>
 - Cython - <http://cython.org/>
- Numba for increased performance - <http://numba.pydata.org/>
- Sphinx to generate local documentation - <http://www.sphinx-doc.org/en/stable/>

- A Python testing framework such as Nose2 or py.test

Input File Formats

The format of the input mapping and bond files is fully described in the documentation at <http://pycgtool.readthedocs.io/>. Sample files used for the atenolol case are provided:

Mapping file, defining bead name, type and component atoms with residue name in square brackets:

```
[36KB]
N1  P1  N1  C12 C13 C14
O1  P1  O3  C1  C2  C3
O2  SP3 O2  C4
C1  SC3 C5  C6
C2  SC3 C8  C9
C3  SC3 C7  C10
N2  P5  C11 O1  N2
```

Bond definition file, listing bond lengths and optionally angles to be measured:

```
[36KB]
N1 O1
O1 O2
O2 C1
O2 C2
C1 C2
C1 C3
C2 C3
C3 N2
```

Simulation Methods

Atomistic - GROMOS

All atomistic simulations performed in the validation of PyCGTOOL, with the exception of the capsaicin reference which was provided by the Sansom group^{S1} and the polyalanine ref-

erence, were performed using the GROMOS 53A6 force field in SPC water using GROMACS 2016. A simulation timestep of 2 fs was used with PME electrostatics and both direct-space electrostatic and Lennard-Jones cutoffs of 1.2 nm with LJ dispersion correction. Constraints were applied to all bond lengths using LINCS. All simulation systems were equilibrated by NVT using the v-rescale thermostat at 310 K for 100 ps, then NPT using the Nose-Hoover thermostat and Parrinello-Rahman barostat at 1 bar for 1 ns. The reference simulation trajectory was collected over 50 ns under the same conditions as the NPT equilibration.

Atomistic - AMBER

The atomistic simulation used as the reference model of polyaniline was performed using the AMBER03 force field in TIP3P water using GROMACS 2016. A simulation timestep of 2 fs was used with PME electrostatics and both direct-space electrostatic and Lennard-Jones cutoffs of 0.8 nm with LJ dispersion correction. Constraints were applied to all bond lengths using LINCS. All simulation systems were equilibrated by NVT using the Berendsen thermostat at 310 K for 100 ps, then NPT using the Nose-Hoover thermostat and Parrinello-Rahman barostat at 1 bar for 1 ns. The reference simulation trajectory was collected over 50 ns under the same conditions as the NPT equilibration.

MARTINI CG

All MARTINI CG models were simulated using GROMACS 2016 using the standard simulation settings provided on the MARTINI website. Electrostatic interactions were calculated using the reaction field method with a short range cutoff of 1.1 nm and screening constant of 15. Lennard-Jones interactions used a potential-shifted cutoff at 1.1 nm. Only bond lengths explicitly listed in the topology as constraints were constrained using LINCS.

Simulation systems were equilibrated first in NVT using the v-rescale thermostat at 310 K for 1 ns with a timestep of 5 fs, then in NPT using the v-rescale thermostat and Berendsen barostat at 1 bar for 100 ns with a timestep of 10 fs. The simulation trajectory was collected

over 500 ns with a timestep of 20 fs using the v-rescale thermostat and Parinello-Rahman barostat at 1 bar.

Validation

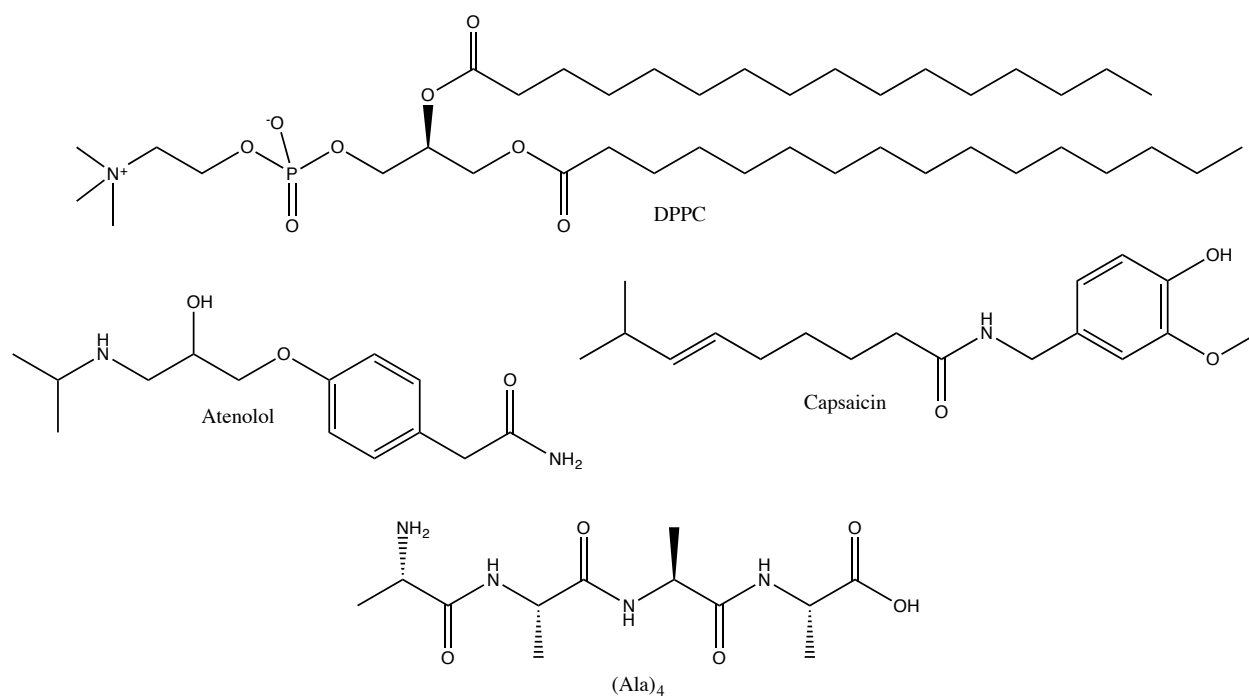


Figure S1: The molecules used for validation of PyCGTOOL.

Lipids

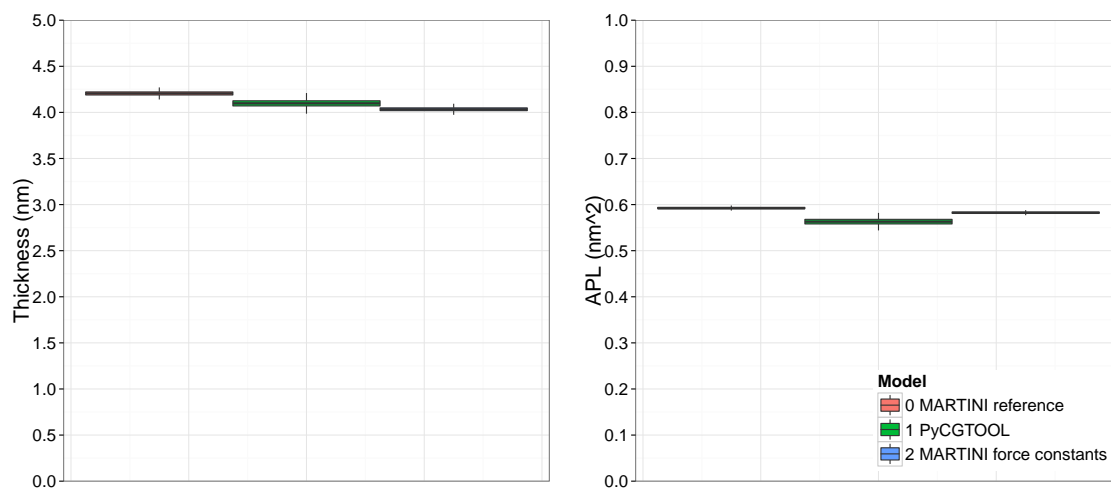


Figure S2: Membrane thickness and area per lipid as measured by RAMSi for the lipid DPPC.

Capsaicin

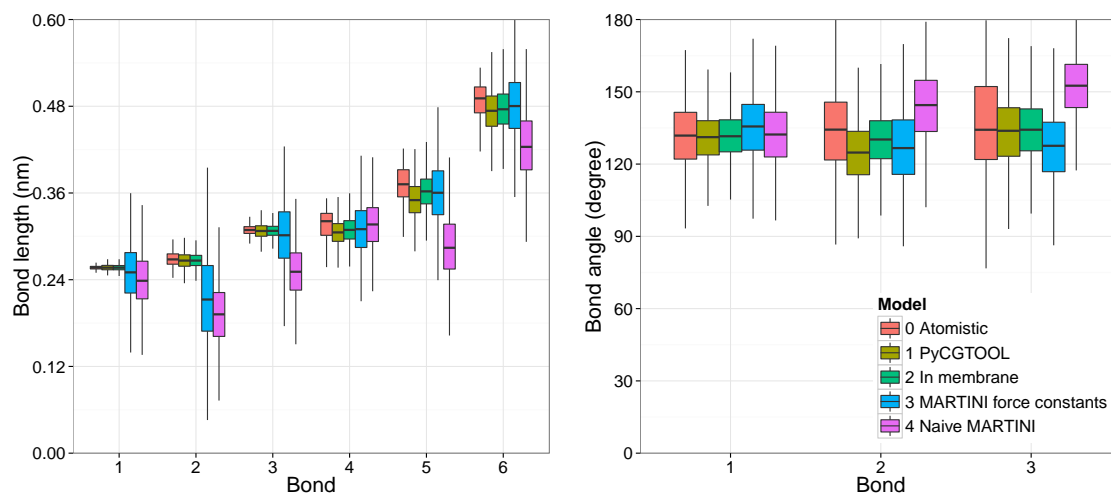


Figure S3: Bond length and angle distributions for the molecule capsaicin using a range of models.

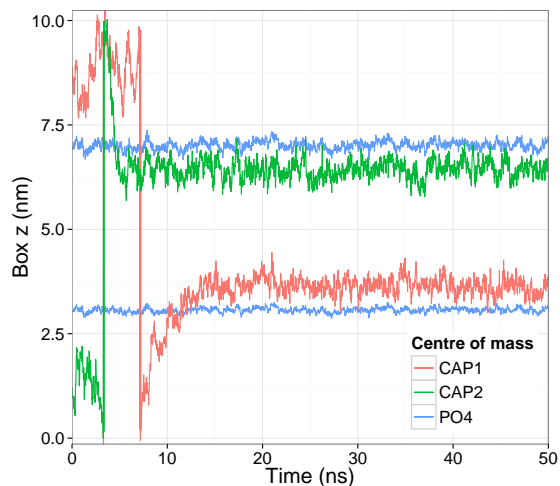


Figure S4: Centre of mass in z profile plot of a capsaicin-popc simulation. Both capsaicin molecules can be seen to enter the membrane within 15 ns where they remain until the end of the 1 μ s simulation. Only the first 50 ns are shown here. The large jumps at approximately 3 ns and 7 ns are artefacts of both molecules crossing the periodic box boundary.

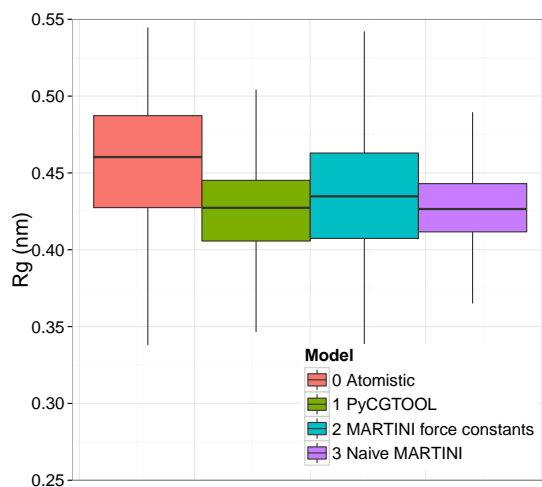


Figure S5: Radius of gyration for capsaicin.

Polymers

For validation of the polymer functionality, a small, four-residue strand of polyalanine was used. A reference simulation was performed using the AMBER03 forcefield^{S2} and the TIP3P water model^{S3} in GROMACS 2016. The standard MARTINI protein mapping was used, whereby an alanine residue is represented by a single ‘P4’ bead. By use of the *output_forcefield*

setting in the advanced options of PyCGTOOL, a full GROMACS forcefield directory can be created. This enables the use of the GROMACS tool *gmx pdb2gmx* to construct the topology of alanine polymers of arbitrary length.

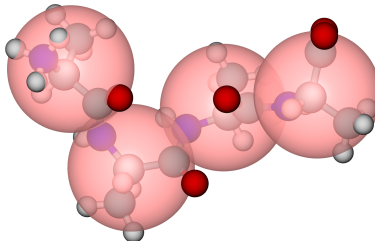


Figure S6: Coarse-grain mapping for the short strand of polyalanine. The atomistic frame was mapped to its pseudo-CG representation for simple visualisation.

Modified Boltzmann Inversion

By applying the transformation eqn 2 to the standardised normal distribution it can be made equivalent to the target functional form, by making the assumption of independent degrees of freedom, and hence no correlation. If this transformation is applied to the standard deviation, the target force constant can be obtained as in eqns 4, 6. GROMACS gives a conversion factor between the harmonic and cos-harmonic forms of the angle potential (eqn 5) which is used to calculate force constants for the cos-harmonic functional form. This method is similar to the Boltzmann Inversion technique,^{S4} but uses the assumption of normality of the measured values to obtain a force constant rather than a tabulated potential.

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \equiv e^{-\frac{k(x-x_0)^2}{2RT}} \quad (1)$$

$$-RT \log f(x) = \frac{RT(x-\mu)^2}{2\sigma^2} \quad (2)$$

$$k(x-x_0)^2 \equiv \frac{RT(x-\mu)^2}{\sigma^2} \quad (3)$$

$$k = \frac{RT}{\sigma^2} \quad \text{simple harmonic} \quad (4)$$

$$k_{\cos} = \frac{k_{\text{harm}}}{\sin^2 x_0} \quad (5)$$

$$k_{\cos} = \frac{RT}{\sigma^2 \sin^2 \mu} \quad \text{cos harmonic} \quad (6)$$

Where x is the internal coordinate, x_0 is the equilibrium value of x , μ is the mean value of x , and σ is the standard deviation of x . The resultant force constants are k for a harmonic functional form, as used for bond lengths, and k_{\cos} for a cosine-harmonic functional form, as used for angles.

Parameters of Generated Models

Table S1: Parameters generated for alternate models of Atenolol.

Bond	PyCGTOOL		Default MARTINI		Naive MARTINI	
	eqm.	f. const.	eqm.	f. const.	eqm.	f. const.
B1	0.370,99	8975.978,46	0.370,99	1250	0.396	1250
B2	0.300,44	12,960.932,05	0.300,44	1250	0.314	1250
B3	0.236,41	-	0.236,41	1250	0.236	1250
B4	0.245,16	-	0.245,16	1250	0.239	1250
B5	0.241,37	-	0.241,37	1250	0.243	1250
B6	0.246,49	-	0.246,49	1250	0.250	1250
B7	0.246,36	-	0.246,36	1250	0.250	1250
B8	0.240,47	-	0.240,47	1250	0.248	1250
A1	109.293,03	17.614,61	109.293,03	25	131.33	25
A2	156.319,64	453.896,88	156.319,64	25	170.96	25
A3	123.849,43	130.752,85	123.849,43	25	118.33	25
A4	122.014,16	4604.874,87	122.014,16	25	120.84	25
A5	118.499,38	4341.751,59	118.499,38	25	119.41	25
A6	128.069,03	68.442,58	128.069,03	25	155.59	25
A7	128.989,10	67.572,42	128.989,10	25	117.80	25

Table S2: Parameters generated for alternate models of Capsaicin.

Bond	PyCGTOOL		Default MARTINI		Naive MARTINI	
	eqm.	f. const.	eqm.	f. const.	eqm.	f. const.
B1	0.256,55	-	0.256,55	1250	0.27	1250
B2	0.268,64	28,104.363,91	0.268,64	1250	0.27	1250
B3	0.308,26	46,366.917,23	0.308,26	1250	0.27	1250
B4	0.316,03	7265.650,83	0.316,03	1250	0.33	1250
B5	0.371,56	4197.408,80	0.371,56	1250	0.38	1250
B6	0.484,86	3014.761,22	0.484,86	1250	0.50	1250
A1	131.471,61	78.363,08	131.471,61	25	120	25
A2	133.328,44	57.540,41	133.328,44	25	147	25
A3	135.969,69	38.406,85	135.969,69	25	168	25

Table S3: Parameters generated for alternate models of DPPC.

Bond	PyCGTOOL		Default MARTINI		Orig. MARTINI	
	eqm.	f. const.	eqm.	f. const.	eqm.	f. const.
B1	0.459,34	1059.133,97	0.459,34	1250	0.47	1250
B2	0.450,33	1069.727,76	0.450,33	1250	0.47	1250
B3	0.347,33	1126.240,56	0.347,33	1250	0.37	1250
B4	0.463,33	1175.279,00	0.463,33	1250	0.47	1250
B5	0.448,89	1116.399,29	0.448,89	1250	0.47	1250
B6	0.447,79	1100.749,21	0.447,79	1250	0.47	1250
B7	0.449,66	1104.576,30	0.449,66	1250	0.47	1250
B8	0.452,05	1133.663,48	0.452,05	1250	0.47	1250
B9	0.448,17	1109.521,26	0.448,17	1250	0.47	1250
B10	0.447,87	1099.023,87	0.447,87	1250	0.47	1250
B11	0.449,94	1105.567,86	0.449,94	1250	0.47	1250
A1	104.165,25	10.370,76	104.165,25	25	-	-
A2	111.600,67	35.233,05	111.600,67	25	120	25
A3	144.842,03	101.121,43	144.842,03	25	180	25
A4	112.859,55	19.330,28	112.859,55	25	-	-
A5	88.028,34	31.882,89	88.028,34	25	-	-
A6	146.602,12	114.459,88	146.602,12	25	180	25
A7	146.611,59	106.839,56	146.611,59	25	180	25
A8	143.950,77	86.925,24	143.950,77	25	180	25
A9	146.617,44	108.750,88	146.617,44	25	180	25
A10	146.188,42	103.189,04	146.188,42	25	180	25
A11	143.421,88	83.924,30	143.421,88	25	180	25

Table S4: Parameters generated for alternate models of Ala4.

Bond	PyCGTOOL		Default MARTINI		Orig. MARTINI	
	eqm.	f. const.	eqm.	f. const.	eqm.	f. const.
B1	0.378,65	27,968.796,45	0.378,65	1250	0.378	1250
A1	118.628,92	31.343,35	118.628,92	25	111.5	25

References

- (S1) Hanson, S. M.; Newstead, S.; Swartz, K. J.; Sansom, M. S. Capsaicin Interaction with TRPV1 Channels in a Lipid Bilayer: Molecular Dynamics Simulation. *Biophys. J.* **2015**, *108*, 1425–1434.
- (S2) Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. *J. Comput. Chem.* **2003**, *24*, 1999–2012.
- (S3) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* **1983**, *79*, 926.
- (S4) Tschöp, W.; Kremer, K.; Batoulis, J.; Bürger, T.; Hahn, O. Simulation of Polymer Melts. I. Coarse-Graining Procedure for Polycarbonates. *Acta Polym.* **1998**, *49*, 61–74.