CHARMM-GUI Polymer Builder for Modeling and Simulation of Synthetic Polymers

Yeol Kyo Choi[1†]*, Sang-Jun Park[1†], Soohyung Park[1], Seonghoon Kim[2], Nathan R. Kern[1], Jumin Lee[1], and Wonpil Im[1]*

[1]Departments of Biological Sciences, Chemistry, Bioengineering, and Computer Science and Engineering, Lehigh University, Bethlehem, Pennsylvania 18015, USA,

[2]School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, Republic of Korea

[†]Both authors contributed equally to this work.

[*]Corresponding authors: choiykyo@gmail.com and wonpil@lehigh.edu

Keywords

polymer modeling, molecular dynamics, coarse-grained simulation, phase separation, self-assembly

ABSTRACT

Molecular modeling and simulations are invaluable tools for polymer science and engineering, which predict physicochemical properties of polymers and provide molecular-level insight into the underlying mechanisms. However, building realistic polymer systems is challenging and requires considerable experience because of great variations in structures as well as length and time scales. This work describes *Polymer Builder* in CHARMM-GUI (http://www.charmm-gui.org/input/polymer), a web-based infrastructure that provides a generalized and automated process to build a relaxed polymer system. *Polymer Builder* not only provides versatile modeling methods to build complex polymer structures, but also generates realistic polymer melt and solution systems through the built-in coarse-grained model and all-atom replacement. The coarse-grained model parameterization is generalized and extensively validated with various experimental data and all-atom simulations. In addition, the capability of *Polymer Builder* for generating relaxed polymer systems is demonstrated by density calculations of 34 homopolymer melt systems, characteristic ratio calculations of 170 homopolymer melt systems, a morphology diagram of poly(styrene-b-methyl methacrylate) block copolymers, and self-assembly behavior of amphiphilic poly(ethylene oxide-b-ethyl ethane) block copolymers in water. We hope that *Polymer Builder* is useful to carry out innovative and novel polymer modeling and simulation research to acquire insight into structures, dynamics, and underlying mechanisms of complex polymer-containing systems.

1. INTRODUCTION

Polymers have been used for a wide range of applications, such as smart surfaces[1], sensors[2], actuators[3], bio-imaging[4], drug delivery[5, 6], and energy devices[7-10], as they are versatile and can be easily mass-produced and processed. To date, owing to technological advances in various fields, more effective and eco-friendly polymer materials are being designed and produced in each application[11, 12]. However, it is challenging to design new polymer-based materials because of the significant time and cost to synthesize prototypes and analyze their physicochemical properties. Furthermore, improving their desired properties and performance requires a good understanding of the relationship between molecular structures and key physicochemical properties.

Computational modeling and simulation have played important roles in polymer science and engineering as they provide molecular-level insight into the underlying mechanisms of macromolecular properties that are difficult to elucidate only with experiments[2, 13, 14]. Current state-of-the-art computational modeling and simulation have been validated enough to interpret experiments and guide new experiments with testable hypotheses. However, as larger spatial scales, longer time scales, and higher levels of realism become possible and necessary, generation of realistic complex polymer systems becomes a major obstacle even for simulation experts. In addition to the reproducibility of simulation outcomes, the grand challenges are how to utilize modeling and simulation techniques effectively to solve practical problems (for experts) and how to lower the high entry barrier in using these models and techniques (for non-experts).

Polymers exhibit interesting and important phenomena over a broad range of length and time scales[15]. To capture this range effectively, most computational polymer models fall coarse-grained (CG) or all-atom models. In the last two decades, CG methods have facilitated the simulation of polymer systems, complementing atomistic simulations and allowing exploration of the behavior of larger systems over longer time scales[16-24]. Several CG models have been developed to reproduce universal (scaling laws) properties of polymers using both bottom-up and top-down approaches for CG force field parameterization.

The bottom-up approaches, which include iterative Boltzmann inversion[18], force matching[25], and inverse Monte Carlo[26], derive parameters from extensive all-atom simulation results. The top-down approaches, such as MARTINI[19-22], SDK[23], or SAFT-γ CGFF[24], use experimental data to optimize force field parameters. The bottom-up approaches can parameterize CG models for specific chemistries, and the top-down approaches are restricted by the number of experimental data available for parameterization. Still, general and systematic approaches are lacking, which makes it a daunting task to model realistic polymer systems. Furthermore, in many cases, atomistic insight is required even if simulations are carried out at a coarser level.

Several programs have been developed to help users build all-atom polymer model systems, including web applications for modeling polymer structures, such as *Polymer Modeler*[27], and stand-alone software packages such as *polymatic*[28] and *pysimm*[29]. *Polymer Modeler* supports 7 pre-built all-atom polymer structures and user-uploaded polymer structures to create polymer systems. *polymatic* and *pysimm* provide methods to prepare cross-linked polymer structure and an API to integrate different features of existing software packages using Python-based scripting, respectively. However, all software requires significant pre-processing to prepare structures, topologies, and parameters of monomers, and initial configurations of desired polymer systems. While expensive commercial programs, such as Material Studio[30] and Schrödinger[31], have state-of-the-art polymer building tools, they also have limitations in terms of system size, and the models from these programs are not transferable to other molecular dynamics (MD) simulation packages.

Most importantly, to investigate physicochemical properties of polymers, it is crucial to analyze the simulation systems from statistical averages on relaxed configurations of polymer chains. However, generating reliable initial configurations of various polymer structures is itself a challenging problem because it requires extensive equilibration simulation. In some cases, an inappropriate initial configuration makes polymer configuration relaxation impossible, even after the simulation is performed beyond

microseconds. Although some molecular packing tools, such as PACKMOL[32], have been used to facilitate modeling of the (known) equilibrated morphology structures, it needs further extensive simulation after modeling initial configurations, and it is impossible to model the structure when its equilibrated morphology is unknown.

To address this shortcoming, we have developed *Polymer Builder* (http://www.charmm-gui.org/input/polymer) in CHARMM-GUI[33-35], which provides a generalized and automated building process to help users build complex polymer systems easily and interactively using a web browser. Given user-specific polymer structure(s) and various system parameters (such as the system size, polymer composition, polymer concentration in solvent, and temperature), *Polymer Builder* performs an equilibration simulation of the polymer structures with the *Polymer Builder* CG model in the web interface. The parameterization method of our CG model is designed to be general and systematic. Relaxed all-atom configurations are generated by converting CG models into corresponding atomistic models, either for direct inspection of the atomistic interaction or for continuation of the simulation with a higher resolution. Importantly, *Polymer Builder* also provides well-validated all-atom simulation inputs for various MD programs, such as CHARMM[36], GROMACS[37], NAMD[38], LAMMPS[39], AMBER[40], GENESIS[41], OpenMM[42], and Desmond[43], allowing users to perform MD simulation with their familiar package(s). The capability and efficiency of *Polymer Builder* are examined by density and characteristic ratio calculations of 34 homopolymer melt systems. Furthermore, its robustness is tested by building and simulating various polymer-containing systems for phase behaviors of block copolymers and self-assembly of block copolymers in aqueous environments, illustrating the versatile simulation contents that one can perform with *Polymer Builder*. Current limitations and future directions of *Polymer Builder* are also discussed briefly.

2. METHODS

2.1 Workflow of *Polymer Builder*

As shown in **Figure 1A**, the overall process to build a polymer system has been generalized and automated in three subsequent steps. Each step is designed to incorporate user's specific options through a web interface and run CHARMM or OpenMM input files. Individual input and output files including generated structures, as well as an archive of all the created files are available in each step. *Polymer Builder* provides a visualization option for the generated system in each step, so that, if necessary, one can go back to the previous step and modify the option interactively. Video demonstrations on how to use *Polymer Builder* are available on the CHARMM-GUI website (http://www.charmm-gui.org/demo/polymer).

Step 1 – Building polymer structure(s)

Currently, *Polymer Builder* provides more than 60 different monomer structures and chemical modifications (**Figure 1B**). *Polymer Builder* adopts a graphical user interface (GUI), allowing users to easily check and design polymer sequences (i.e., monomer type, polymer length, and capping of polymer ends). One can build a polymer structure with almost all combinations of the monomer structures and by specifying the repeating number of each monomer (**Figure 1C**). Such a flexible user interface enables *Polymer Builder* to cover diverse polymer structures. Detailed procedures on how to build polymer structures using *Polymer Builder* are described in Supporting Tutorial and video demonstrations.

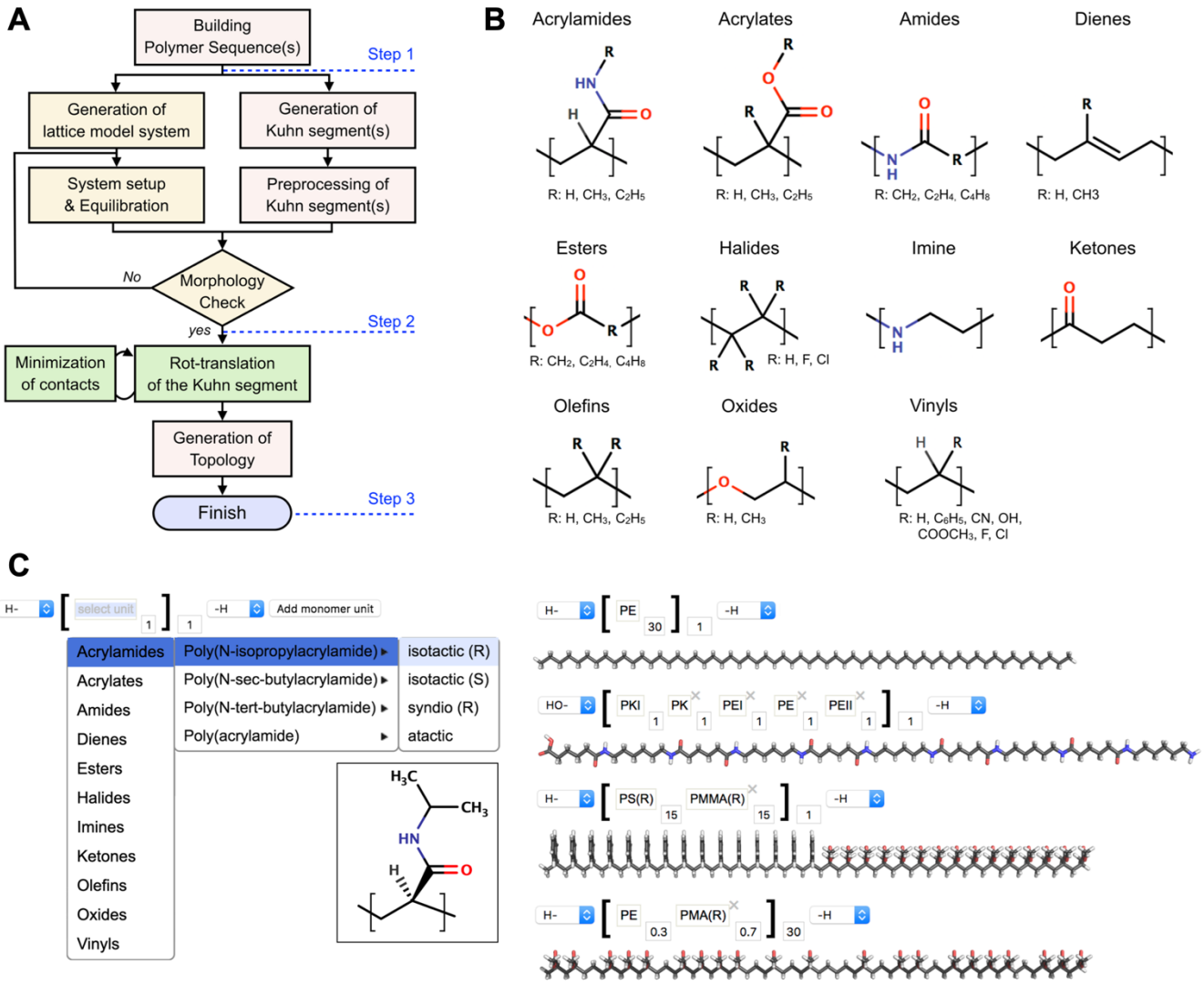

Figure 1. (A) Workflow of *Polymer Builder*. Steps involving CG and all-atom models are in yellow and red, respectively, and those involving both resolutions are in green. (B) Monomer classes available in *Polymer Builder* and corresponding chemical structures. (C) Illustrative snapshots of *Polymer Builder* user interface. (left) One can build a polymer sequence using one of the predefined monomers whose chemical structure is displayed on the side panel. (right) Illustrative snapshots and corresponding output structures are shown for polyethylene, nylon 66, poly(styrene-co-methyl methacrylate) block copolymer, and random copolymer of polyethylene and polymethacrylate. Carbon, oxygen, nitrogen, and hydrogen atoms are colored in black, red, blue, and white, respectively.

Step 2 – Generation of a relaxed system using *Polymer Builder* CG model

Polymer Builder not only provides single polymer chain modeling capability, but also supports a "melt system" option for modeling amorphous and molten states and a "solution system" option for modeling polymers dissolved in various solvents. **Figure 2** illustrates how *Polymer Builder* operates to generate a relaxed system. Each polymer structure from Step 1 is first fragmentized using an estimated (single) CG bead size based on the monomer units, where different fragment structures (in a block copolymer) are modeled as different CG bead types with different interaction parameters, which is elaborated below. *Polymer Builder* then performs two tasks simultaneously. The first one is to model all-atom library structures for each CG bead type by considering its connecting orientation to neighboring ones (**Figure 2A**), and these all-atom structures are used to replace each CG bead in Step 3. The second task is to run a CG simulation to equilibrate a melt or solution system using OpenMM[42]. The *Polymer Builder* CG force field was developed by combining solubility parameters and machine learning techniques, and the force field parameters (i.e., bead sizes and interaction parameters) have been extensively validated with various experimental data and all-atom simulations (see next section). Note that a CG lattice structure is first built based on the force field parameters, and the CG equilibration simulation is then performed longer than an estimated relaxation time of the selected polymer structure (**Figure 2B**).

Step 3 – Generation of an all-atom simulation system and inputs

The CG beads in the relaxed configurations (melt or solution systems) from Step 2 are replaced by all-atom segments in the library built in Step 2 via its translation and rotation to minimize bad contacts (**Figure 2C**). After the replacement, users can obtain a relaxed all-atom polymer system (**Figure 2D**) with necessary topology, force field parameter, and simulation input files for further simulations using any of supported simulation programs such as CHARMM, GROMACS, NAMD, LAMMPS, AMBER, GENESIS, OpenMM, and Desmond.

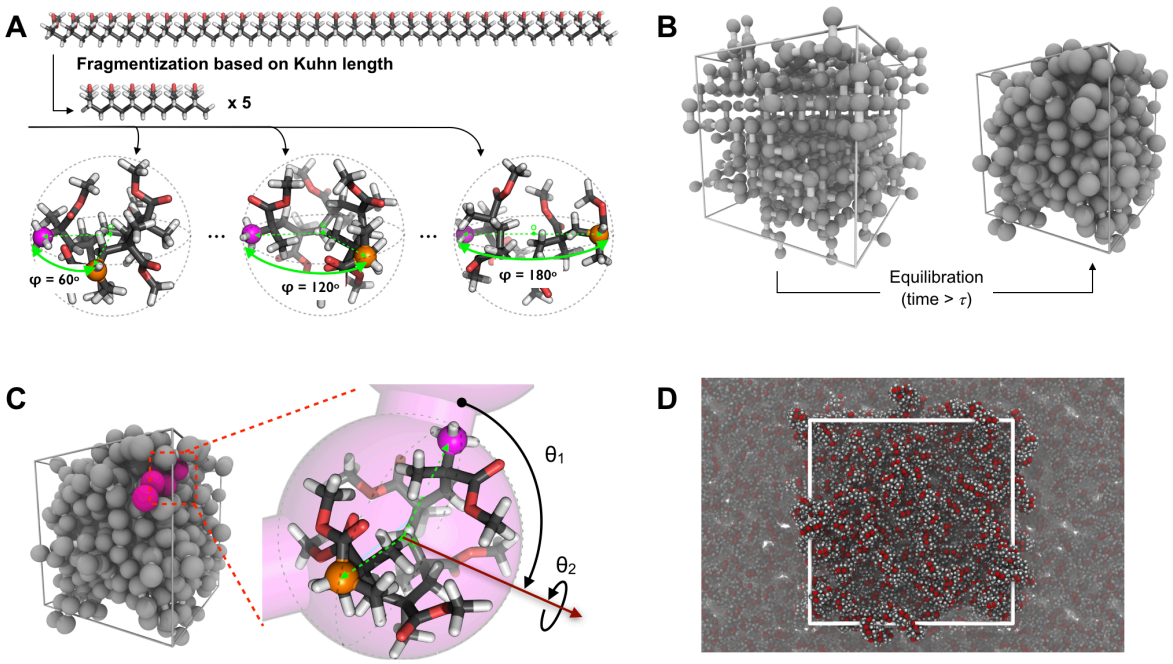

Figure 2. Snapshots to illustrate how *Polymer Builder* operates. (A) A $PMMA_{30}$ (polymethyl methacrylate with 30 monomer units) chemical structure and an initial process to make Kuhn fragment library. $PMMA_{30}$ is fragmented based on its Kuhn length. The Kuhn length of PMMA is 15.3 Å that corresponds to six monomers per fragment and $PMMA_{30}$ can be divided into five identical Kuhn fragments. Spherical and positional restraints are applied to the Kuhn fragment to fit this all-atom structure to a corresponding CG bead. The positional restraints are applied to make the angle between three points (the center of mass of the Kuhn fragment and the positions of two carbon atoms that are connected to other fragments) ranging from 60° to 180° by 10° (see Supporting Tutorials for the details). The connecting carbons are marked with magenta and orange spheres. (B) Snapshots of initial and final simulation systems of 50 $PMMA_{30}$ using the *Polymer Builder* CG model. τ is the (longest) relaxation time of the CG polymer model. (C) Replacement of each CG bead with the corresponding all-atom structure from the structure library generated in (A) with rotation and translation optimization to minimize bad contacts. (D) Final all-atom model obtained for a melt system of 50 $PMMA_{30}$ with the primary simulation system in a white box and surrounding image systems in the XY directions. Carbon, oxygen, and hydrogen atoms are colored in black, red, and white, respectively.

2.2 *Polymer Builder* CG Model

A bead-spring model (also commonly known as Kremer-Grest models)[44, 45] is employed to represent the constituent polymers. All bonded interactions between CG beads are modeled using a harmonic potential $U_{bond}(r) = k_{bond} (r - r_0)^2$ with $k_{bond} = 100\ k_B T/\sigma^2$ and $r_0 = \sigma$, where $k_B T$ is a thermal energy and σ is a bead diameter. The nonbonded interactions between all bead pairs are modeled through the combination of the 12-6 Lennard-Jones potential and the Weeks-Chandler-Andersen potential.[46] Developments of *Polymer Builder* CG model parameters are subsequently described in detail.

2.2.1 Choice of σ

Developing a CG force field requires a specific mapping method about how many all-atom monomers are mapped onto a single CG bead (e.g., **Figure 2A**). For polymers, the spatial correlations between monomers become negligible beyond a characteristic length scale known as the Kuhn length (b)[47]. At a length scale above b, the chain can be treated as a fully flexible polymer consisting of N_b Kuhn monomers. The Kuhn length of a polymer is defined by the ratio of the mean square end-to-end distance ($\langle h^2 \rangle$) and the fully extended length of a polymer (h_{max}), $b = \langle h^2 \rangle / h_{max}$. Flory defined the characteristic ratio (C_∞) as the ratio of the actual unperturbed $\langle h^2 \rangle$ and that of a freely jointed chain,[47]

$$C_\infty = \langle h^2 \rangle / (n \times l^2) \quad (1)$$

where l is an average backbone bond length and n is the number of backbone bonds in a polymer chain. Using this definition and $h_{max} = n\ l \cos(\theta/2)$ where θ is an average backbone bond angle, the Kuhn length can be rewritten as[47]

$$b = C_\infty l / \cos(\theta/2) \quad (2)$$

The Kuhn length provides crucial information about the number of monomers that can be represented by a single CG bead. To obtain C_∞ for each monomer unit in *Polymer Builder*, we performed all-atom MD simulations of homopolymers of each monomer with N_m = 10, 20, 30, 40, and 50 in a melt system of 10 ×

10 × 10 nm^3. Then, $\langle h^2 \rangle$ was fitted with Equation (1) and **Figure S1** shows the results for four representative polymers. **Figure 3** shows the comparison of C_∞ obtained from all-atom simulations with experimental data for various homopolymers, indicating that the simulation results agree well with experiments.

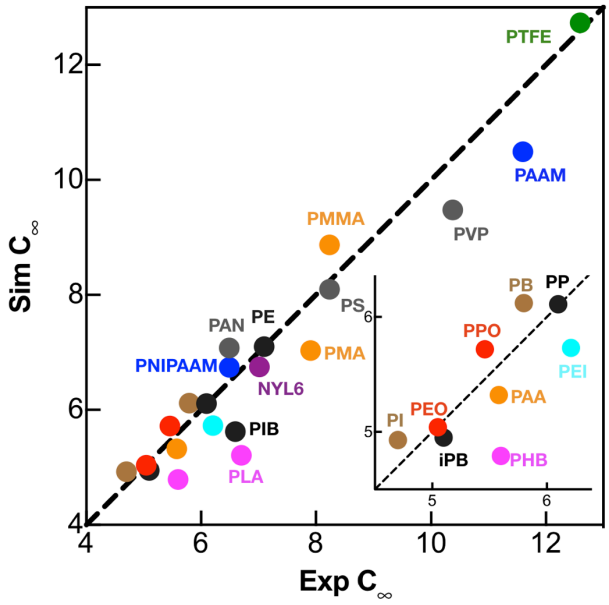

Figure 3. Comparison of simulated and experimental characteristic ratios (C_∞) of various homopolymers. Some experimental characteristic ratios are derived from the persistence length and Kuhn length (see **Table S1** for polymer full names).

Next, to investigate a spatial correlation in our bead-spring model, we performed CG simulations of homopolymers with N_{CG} = 10, 20, 30, 40, and 50 in a melt system of 10 × 10 × 10 nm^3 with a range of σ from 7 to 13 Å (covering all polymer units in *Polymer Builder*). Using Equation (1), fitting $\langle h^2 \rangle_{CG} / \sigma^2$ to $N_{CG} - 1$ yields $C_\infty = 1.48$ (**Figure S2**), indicating a mild spatial correlation (as expected since $C_\infty = 1$ can be obtained only for the ideal freely-jointed chains). This in turn leads to a relation $b = 1.48\sigma$ from Equation (2), so that a bead diameter σ of a homopolymer can be directly obtained from its Kuhn length, b.

To generalize this method for copolymer structures, we calculated the number of all-atom monomers representing a Kuhn segment of a homopolymer ($N_{m:b}$) and corresponding number of all-atom monomers in a CG bead ($N_{m:\sigma}$) using $N_{m:\sigma} = \sigma\, N_{m:b} / b = N_{m:b} / 1.48$ (**Table S1**). We assume that each monomer contributes an amount of $1 / N_{m:\sigma}$ to the formation of a copolymer CG bead, so that we can sequentially determine bead segments ($S^{(k\text{-}j)}$) where j is the starting monomer number and k is the first monomer number that makes $\sum_{i=j}^{j+k} 1/N_{m:\sigma}^{(i)}$ greater than 1. We can then approximate each bead segment to a sphere with a diameter corresponding to $\sigma^j = (6/\pi \sum_{i=j}^{j+k} V_m^{(i)})^{1/3}$, where V_m is the monomer volume and can easily be determined from the monomer structure. The resulting σ^j are different for different segments with different monomer units in a copolymer. Note that the *Polymer Builder* CG model does not use these different diameters directly as it is practically impossible to parameterize associated interaction parameters for any user-specified copolymer. Instead, the maximum diameter (σ_{max}) among the different bead sizes is used to estimate the number of CG beads (N_{CG}) in the polymer by $N_{CG} = \text{int}\,(6/\pi \sum_{i=1}^{N_m} V_m^{(i)} / (\sigma_{max})^3)$. Then, a diameter of N_{CG} beads can be calculated by $\sigma = (6/\pi \sum_{i=1}^{N_m} V_m^{(i)} / N_{CG})^{1/3}$; n.b., σ is always a bit larger than σ_{max}. Using σ and N_{CG}, from the first CG bead, we can sequentially assign the monomer units by adding its V_m until the sum is close to $\pi\sigma^3 / 6$. In this way, depending on different monomer sizes in a polymer, the number of monomer units in a CG bead ($N_{m:\sigma}$) is different in each bead and thus capture V_m of different monomer types with a single σ and its associated interaction parameters.

2.2.2 Evaluation of χ from Solubility Parameters

With σ for a given polymer in hand, the next task is to determine the interaction parameter. The Flory-Huggins parameter χ describes the degree of segregation in polymer blends and polymer solubility in various solvents, which can be used as an intermediate parameter connecting the microscale (all-atom) model and the mesoscale (CG) model. Based on a simple idea that two materials with similar solubility

parameters have balanced forces and thus are miscible, the Hildebrand solubility parameter concept has been widely used to estimate χ.[48, 49] In the regular solution theory, the solubility parameter is formally defined as $\delta^2 = \Delta E_{vap} / V_{molar}$, where ΔE_{vap} is the molar energy of vaporization and V_{molar} is a molar volume of the substance. χ can be calculated by $\chi = V_r (\delta_{t,A} - \delta_{t,B})^2 / k_B T + \beta$, where β is an empirical constant (β = 0 for polymer-polymer interactions and $\beta = 0.34$ for polymer-solution interactions), $\delta_{t,A}$ and $\delta_{t,B}$ are the total solubility parameters of two compounds A and B, and V_r is the geometric mean of the polymer segment molar volumes for polymer-polymer interactions or the solvent molar volume for solution systems.[50] Therefore, molecular mixing of two components can occur when χ is close to zero. However, derivation of χ for the compounds having polar and hydrogen-bonding interactions by above equations usually yield poor results. To address this shortcoming, Hansen divided Hilderbrand solubility parameters into contributions from dispersion forces (d), polar forces (p), and hydrogen bonding effects (h), i.e. $\delta_t = (\delta_d^2 + \delta_p^2 + \delta_h^2)^{1/2}$ and suggested[50]

$$\chi = 0.6\, V_r ((\delta_{d,A} - \delta_{d,B})^2 + 0.25\, (\delta_{p,A} - \delta_{p,B})^2 + 0.25\, (\delta_{h,A} - \delta_{h,B})^2) / k_B T \quad (3)$$

By design, this formula shows very good estimation of χ for systems having strong polar or hydrogen bonding interactions. To estimate χ for an arbitrary polymer system, the group contribution methods for estimating δ, which are based on the knowledge of structural fragments within the molecule, are by far the most popular way of calculating δ. In the *Polymer Builder* CG model, the group contribution method of Stefanis and Panayiotou[51] is used to estimate the Hansen solubility parameter of all supported monomer units (**Table S1**). These solubility parameters are used to calculate χ, and then χ is used to obtain cross-interaction parameters (ϵ) between different CG bead types. Because of different nature of bead-bead interactions for polymer-polymer and polymer-solvent due to its size, we have developed the separate schemes to parameterize ϵ for melt and solution systems.

2.2.3 Choice of ϵ for Melt Systems

To establish interaction strengths between CG beads in melt systems, we start from Lennard-Jones (LJ) fluid whose reported reduced critical temperature is $T_c^* = k_B T_c / \epsilon = 1.31$ in the case of the untruncated LJ potential[52]. We note that the critical temperature (T_c) of the constituent monomer groups are readily available within the framework of the group contribution theory[53], from which the self-interaction strengths between the same CG bead type A is given by $\epsilon_{AA} = k_B T_{c,A} / 1.31$. Even though it is common to use the Lorentz-Berthelot mixing rules for interactions between different species, i.e., $\epsilon_{AB} = (\epsilon_{AA}\ \epsilon_{BB})^{1/2}$, polymer blend systems are known to be highly sensitive to ϵ_{AB}, and thus it is necessary to use a different approach for accurate estimation of ϵ_{AB}[54]. In the *Polymer Builder* CG model, ϵ_{AB} in melt systems was derived from a machine learning method using ϵ_{AA}, ϵ_{BB}, T, and χ as the features, as there is no simple way to derive ϵ_{AB} from ϵ_{AA}, ϵ_{BB}, T, and χ.

First, to establish a relationship between χ and T for given ϵ_{AA}, ϵ_{BB}, and ϵ_{AB}, we follow the simulation approach of Groot et al[55]. Briefly, the system consists of a biphasic mixture of A and B. After equilibration MD simulation, the two phases mix to a certain degree, which leads to the free energy, $F / k_B T = \phi_A \ln\phi_A / N_A + \phi_B \ln\phi_B / N_B + \chi^* \phi_A \phi_B$, where ϕ_A and ϕ_B are the volume fractions of A and B molecules in the B and A regions, and N_A and N_B are the number of beads per A and B molecules ($N_A = N_B = 1$ for our binary mixture simulations). Note that we use χ^* (calculated from simulations with a set of T, ϵ_{AA}, ϵ_{BB}, and ϵ_{AB}) to distinguish it from χ (calculated from chemical structures) in Equation (3). The minimum free energy at $\partial F/\partial\phi_A = 0$ leads to

$$\chi^* = \ln[(1 - \phi_A) / \phi_A] / (1 - 2\phi_A) \quad (4)$$

Using Equation (4), we calculated χ^* as a function of T. As shown in **Figure 4A**, the correlation between χ^* and T can be approximated by a linear relationship, $\chi^* = \alpha/T + \beta$, where α and β correspond to the enthalpic and entropic contributions to χ^*, respectively.[50] To estimate χ^* for different sets of T, ϵ_{AA}, ϵ_{BB}, and ϵ_{AB}, we simulated 820 bi-phase systems with $\epsilon_{AA} = k_B T$ and $\epsilon_{AA} \geq \epsilon_{BB} \geq \epsilon_{AB} \geq 0.5\ k_B T$ at a temperature

range of $\epsilon_{BB}/k_B \geq T \geq 0.5\ \epsilon_{AA}/k_B$, where ϵ_{BB}, ϵ_{AB}, and T were decreased in a step of 0.05 k_BT. Then, we trained a machine to output ϵ_{AB} using ϵ_{AA}, ϵ_{BB}, T, and χ^* as the features.

Kernel ridge regression (KRR) is one of the most popular supervised learning approaches, which has been used for prediction of molecular properties in several studies[56, 57]. The basic idea of KRR is to map input features into a higher-dimensional space, where a linear relation between the transformed features and the property of interest could be established. In this study, KRR was performed with scikit learn package[58] with default parameters, and 655 data (80% of 820 data set) was randomly chosen as a training set and the rest was used as a validation set. **Figure 4B** shows the target and predicted values for ϵ_{AB}, which manifests an excellent prediction with $R^2 = 0.987$. **Figure 4C** summarizes a workflow describing the overall process for parameterization in melt systems. Detailed procedures on how to calculate interaction parameters are described in Supporting Tutorial 3 using an example.

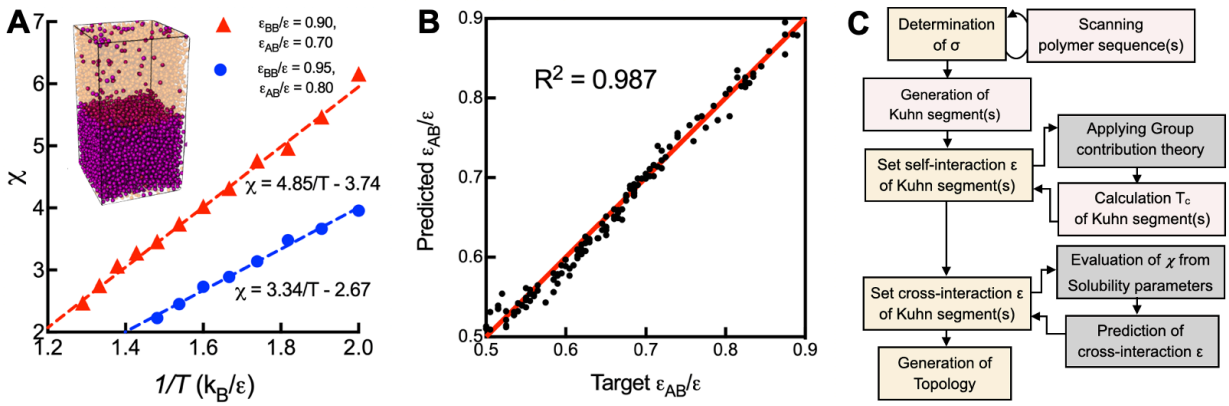

Figure 4. (A) Flory-Huggins parameter χ as a function of inverse temperature for a binary mixture of CG beads. The dashed lines are the linear fits to the simulation data. (B) Target and predicted values for ϵ_{AB}. (C) Workflow of CG parameterization process in melt systems. Steps involving CG and all-atom models are in yellow and red, respectively, and calculations related to the group contribution theory and prediction of ε_{AB} are in gray.

2.2.4 Choice of ϵ for Solution Systems

In general, the ε parameterization method used for melt systems cannot be applied for solution systems because of small solvent molecules (for which Kuhn length cannot be defined). To address this issue, we first set ϵ to $k_B T_c$ / 1.31, where T_c is the critical temperature of a specific solvent. Then, we employed a commonly used approach for polymer solutions[59], where a potential energy function (for different bead type interactions) is defined as the combination of LJ and Weeks-Chandler-Andersen (WCA) potentials with a tuning parameter (λ),

$$V(r) = (1-\lambda)\, V_{LJ}(r) + \lambda\, V_{WCA}(r) \quad (5)$$

λ controls the interaction strength from attractive ($\lambda = 0$) to repulsive ($\lambda = 1$) interactions. Since there was no available quantitative relation between λ and χ, we derived one in this work using the Flory exponent (v) (that describes the scaling of the chain size (or radius of gyration, R_g) as a function of molecular weight) as an intermediate parameter connecting the microscale (all-atom, χ) model and the mesoscale (CG, λ) model. First, the relationship between χ and v from the all-atom model was obtained from single chain simulations of various homopolymers with different molecular weights (N_m = 10, 20, 30, 40, and 50) in 10 different solvents that have various χ values ranging from 0.04 to 1.88 (**Table S2**). **Figure 5A** shows R_g for PP and PVA as a function of molecular weight (M_w) in cyclohexane and acetone at 300 K, respectively. Fitting the data to the power law yields v = 0.61 (PP in cyclohexane), 0.47 (PP in acetone), and 0.32 (PVA in cyclohexane), for which the corresponding χ values are 0.04, 0.46, and 1.87. This indicates that v and χ are anti-correlated. All v values obtained from MD simulations as a function of χ are shown in **Figure 5B**. In the framework of Flory theory, v values are 0.6, 0.5, and 0.33 when a polymer dissolves in good, theta, and poor solvents, respectively.[47] Therefore, in the fitting, v = 0.6 and v = 0.33 were used as upper and lower bounds and the result shows an excellent agreement with the theory (R^2 = 0.97).

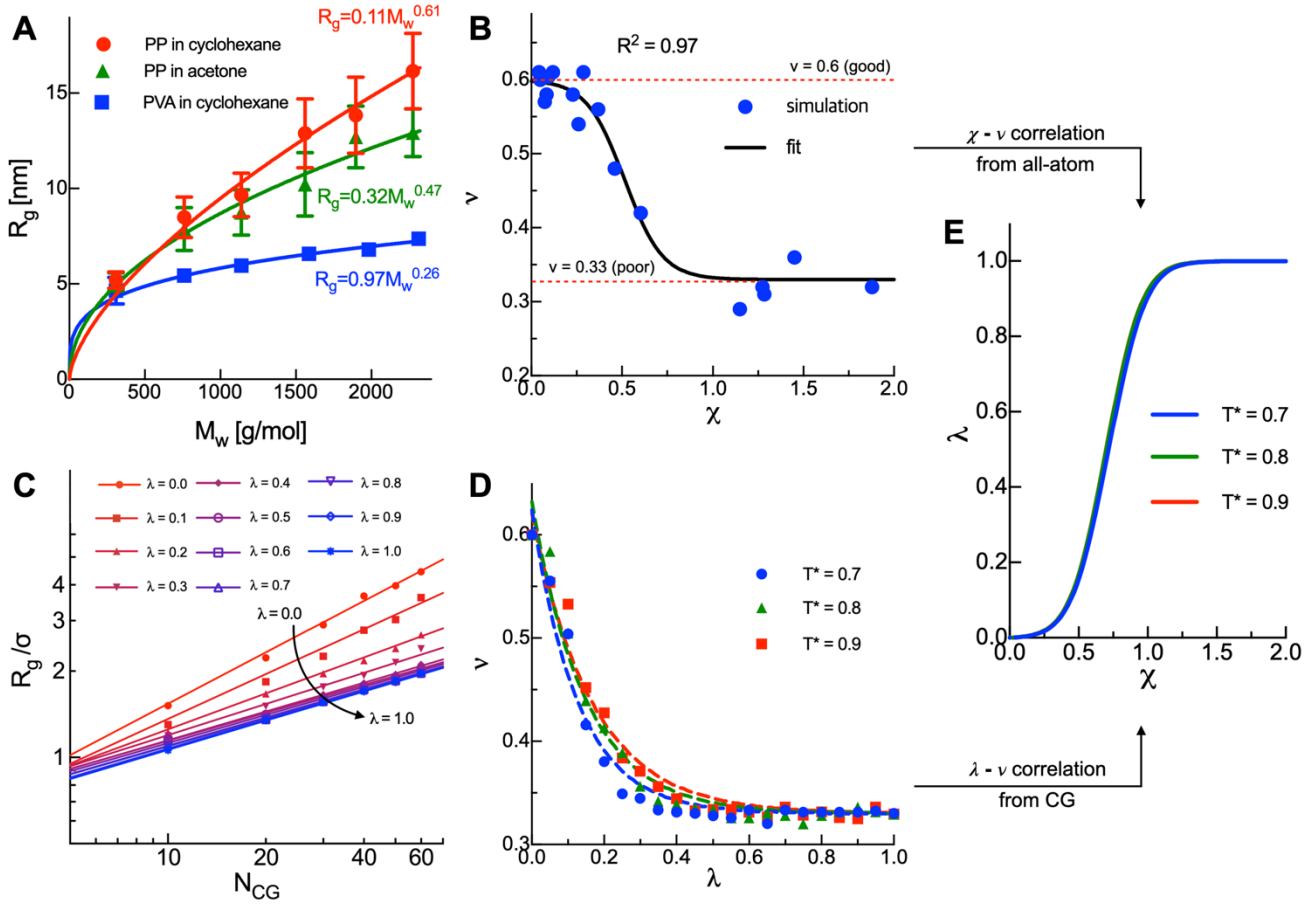

Figure 5. (A) Radius of gyration of PP and PVA as a function of molecular weight (M_w) in different solvents. The solid lines are the fits to the corresponding all-atom simulation data points. (B) Flory exponent v (i.e., v in $R_g \propto M_w^v$) as a function of Flory-Huggins interaction parameter χ for PP, PVA, PMA, PNIPAAM, PE, PLA, and PAN in various solvents (blue circle) and fit (solid line). Dashed lines are the upper and lower bounds reported in the theory and experiments. (C) Radius of gyration of CG polymers as a function of number of CG beads (N_{CG}) and the tuning parameter (λ) that controls the solvent quality from good (λ=0) to poor (λ=1). (D) Flory exponent v as a function of λ (circle) and fit (dashed line) at various reduced temperature. (E) λ as a function of χ at various reduced temperatures.

Then, the relationship between v and λ for the *Polymer Builder* CG model was obtained from simulations of a single chain polymer with different chain lengths (from 2 to 18 beads) in solvent where λ was changed

from 0 to 1 with 0.05 interval. **Figure 5C** shows the logarithmic plots of R_g / σ as a function of number of CG beads (N_{CG}) with various λ. As λ increases, the slope of R_g / σ gradually decreases from 0.64 to 0.41, indicating that adjusting λ can controls long range structural properties of CG polymers. We normalized v based on the upper and lower bounds (0.6 to 0.33), and fitted v to λ at reduced temperatures T^* = 0.7, 0.8, and 0.9 (**Figure 5D**). Combining both χ-v (**Figure 5B**) and λ-v in (**Figure 5D**) relationships, we obtained a desired quantitative relationship between χ and λ (**Figure 5E)**. Therefore, the interaction energy for different bead types (polymer-polymer and polymer-solvent) in solution systems can be calculated using Equation (5).

2.3 *Polymer Builder* CG Equilibration

After the construction of a lattice model and derivation of CG parameters for a polymer system in a melt or solution state, *Polymer Builder* performs CG equilibration to generate relaxed polymer configurations (**Figure 2B**). Therefore, it is crucial to estimate equilibration time and provides the expectation time to users as a web-based cyberinfrastructure. For most polymer systems, relaxation of chain configurations is slow compared to the density or pressure, so a chain-level measurement is required to assess whether the system reaches equilibration. To this end, we generated the systems consisting of 100 polymer chains with N_{CG} from 9 to 72 beads, and the end-to-end vector autocorrelation function ($\langle \mathbf{u}(t)\cdot\mathbf{u}(0)\rangle$) was analyzed as a function of N_{CG} at various temperature (**Figure S3**). At T^* = 0.5, $\langle \mathbf{u}(t)\cdot\mathbf{u}(0)\rangle$ decays to zero only for N_{CG} = 9 and 18 by the end of 250 ns, indicating an insufficient relaxation of the polymer chains with N_{CG} = 27, 36, and 72. To estimate the relaxation time, we fitted the data using the Kolrausch-Williams-Watts stretched exponential,[60] $G(t) = \exp[-(t/\tau)^{\beta}]$ with $0 < \beta \leq 1$. The calculated average relaxation time, $\langle\tau\rangle = \int G(t)dt = \tau\Gamma(1/\beta)/\beta$, ranges from 7.9 to 810.9 ns, when N_{CG} increases from 9 to 72 beads. **Figures S3B** and **S3C** show that $\langle \mathbf{u}(t)\cdot\mathbf{u}(0)\rangle$ decays to zero within 250 ns even for N_{CG} = 72 at T^* = 0.6 and 0.7. A logarithmic relationship between N_{CG} and $\langle\tau\rangle$ at various temperature show a linear relationship and provide a

semiempirical relation to estimate the relaxation time of polymer systems (**Figure S3D**), which is used in *Polymer Builder*.

2.4 Simulation Details

All MD simulations were performed with Gromacs 2018.1 for CG and all-atom MD. To control the temperature, a Nosé-Hoover temperature coupling method[61] was used with τ_t = 1 ps. The pressure was maintained at 1 bar using the Berendsen and Parrinello-Rahman barostats[62] for the equilibrium and production run with τ_p = 5 ps and a compressibility of 4.5 x 10^{-5} bar^{-1}, respectively. Neighbor lists were built using the Verlet cut-off scheme with a cut-off radius of 3σ and 1.2 nm for the CG and all-atom MD, respectively. The linear constraint solver (LINCS) algorithm[63] was used to constrain the hydrogen bond length. All simulations were conducted using a leap-frog integrator with time-steps of 50 and 2 fs for CG and all-atom MD, respectively. The CG model dynamics is faster because the CG interactions are much smoother compared to atomistic interactions. On the basis of comparison of diffusion constants in the CG models and in all-atom models, the effective time sampled using CG is 6 - 10 times longer. When interpreting the simulation results with the CG model, the conversion factor is 8, which is the effective speed up factor in the diffusional dynamics of CG water compared to the real water. Electrostatic interactions were calculated using particle mesh Ewald[64] with a cutoff of 1.2 nm in all-atom MD. The all-atom force field parameters in *Polymer Builder* are based on the CHARMM Generalized Force Field (CGenFF).[65] To cover almost all combinations of monomer units, we have prepared 66 residues (monomer units) and 1,004 patches (connecting monomers). Note that no specific optimization has been performed. Detailed procedures on how to use *Polymer Builder* are described in Supporting Tutorial using four example systems: i) single polymer chains generation, ii) self-assembly of PEO_{76}-PPO_{29}-PEO_{76} block copolymer in water, iii) a ternary blend system of a block copolymer ($P(S_{180}$-b-$MMA_{220})$) and two homopolymers (PS_{180} and $PMMA_{220}$), and iv) PET_{50} membrane with CO_2 molecules.

3. RESULTS AND DISCUSSION

3.1 Generation of Relaxed Polymer Systems

To illustrate the capability of *Polymer Builder*, we investigated the homopolymer melt systems, phase separation of PS-b-PMMA block copolymer, and self-assembly behavior of amphiphilic PEO-b-PEE block copolymer in solution. Building relaxed polymer systems is crucial to predicting the properties of these materials. In particular, reliable polymer configurations are essential to optimize polymer mechanical properties, transport phenomena of small molecules through a polymer membrane, and surface and interface interactions in polymer composite materials. These properties impact polymer's performance in applications such as separation processes, packaging, and drug delivery[66, 67]. To check the quality of polymer structures from *Polymer Builder*, we prepared an initial configuration of 250 PE_{50} chains from *Polymer Builder*, as well as a stack of linear chains (**Figure 6A**) and compared the end-to-end distance distributions of PE_{50} with theoretical estimation at 440 K; n.b., the melting temperature (T_m) of PE is about 420 K. As expected, the initial polymer system generated by simple stacking takes long to make relaxed configurations (**Figure 6B)**, where the end-to-end distance distribution deviates significantly from the theoretical distribution even after 50-ns equilibration. The situation would deteriorate for longer polymers (i.e., long relaxation time) and it may not be possible to obtain a relaxed configuration even after extensive equilibration. On the contrary, the system generated from *Polymer Builder* shows a distribution close to theory only after 10-ns equilibration (**Figure 6B**). To further investigate the formation of relaxed polymer configurations of high molecular weight polymers, initial configurations of 217 PE_{200}, 144 PE_{300}, 108 PE_{400}, and 86 PE_{500} were prepared using *Polymer Builder*, and the end-to-end distance probability distributions were compared with theoretical estimations after 10-ns equilibration at 450 K (**Figure 6C**); n.b., a short 10-ns simulation was performed just to illustrate the quality of the initial relaxed structure from *Polymer Builder*. The distributions obtained from all-atom simulations show excellent agreements with

corresponding theoretical estimations. Furthermore, highly entangled polymers were observed in all systems (**Figure S4**). This indicates that *Polymer Builder* can generate relaxed initial configurations of polymer systems regardless of the polymer length.

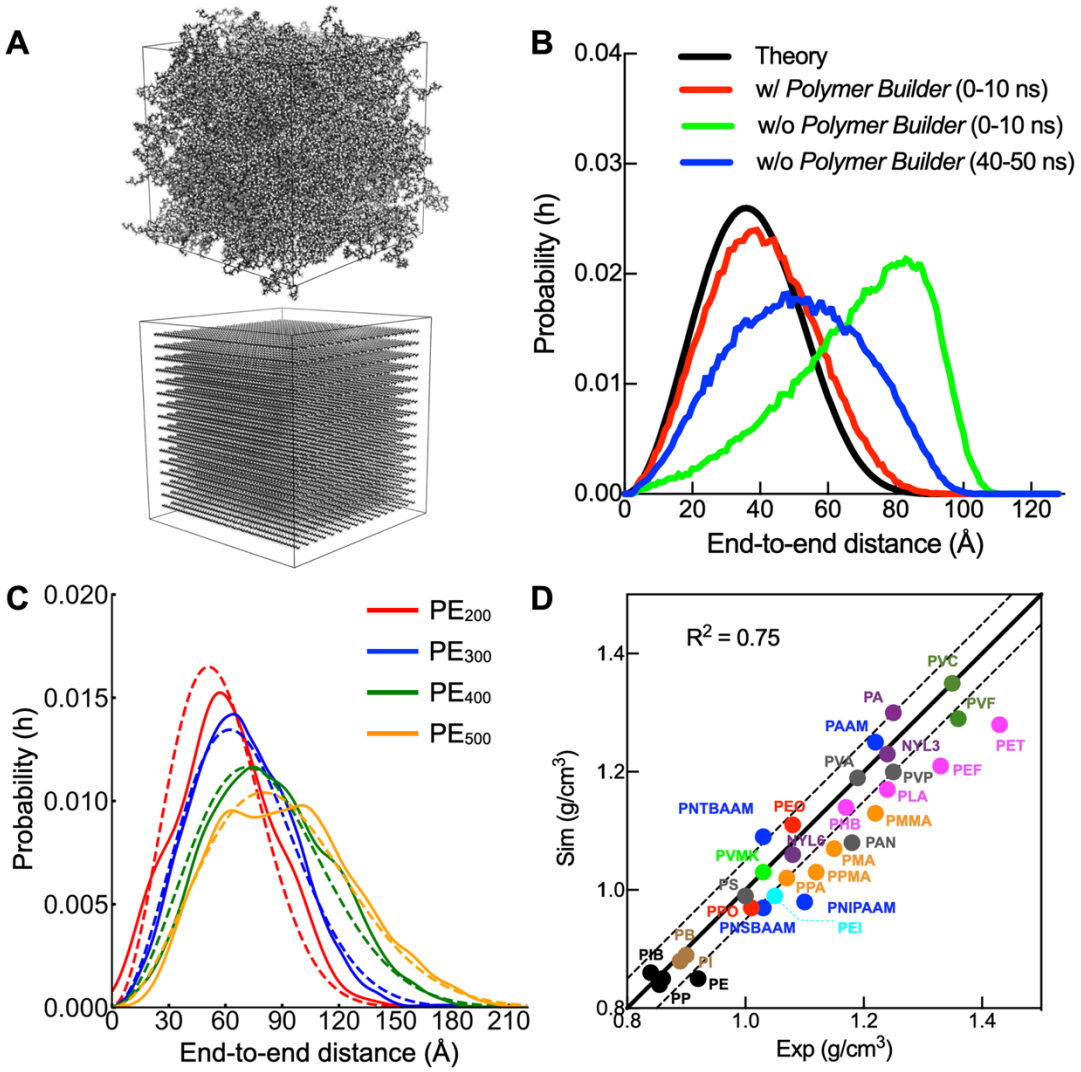

Figure 6. (A) Initial all-atom configuration of 250 PE_{50} chains from *Polymer Builder* (upper) and a stack of linear chains (lower). (B) Probability distributions of PE_{50} end-to-end distance from the initial configurations of (A) and a worm-like chain model (black line). (C) Probability distributions of end-to-end distances of 217 PE_{200}, 144 PE_{300}, 108 PE_{400}, and 86 PE_{500} from 10-ns all-atom simulations at 450 K (solid line) and wormlike chain model (dashed line). (D) Comparison of simulation and experimental densities of

34 homopolymers at 1 bar and 298.15 K. The straight line corresponds to the perfect agreement and the dashed lines delineate the 95 % confidence area. The polymers belonging to the same class are colored together (see **Table S1** for polymer full names).

To investigate bulk properties as well as long-range structural properties of polymers, we have performed simulations of 34 homopolymer melt systems. The relaxed configurations that contained 100 homopolymer chains with 50 monomer units per chain were generated using *Polymer Builder*. Each initial configuration underwent a simulated annealing stage that began at $T = T_m + 50$ K and ended at $T = 298.15$ K with a cooling rate of 0.01 K / ps to prevent formation of defects that could be generated by a sudden temperature change. **Figure 6D** shows the comparison of the densities obtained from the simulations with the experimental data. For most homopolymers, the experimental densities are well reproduced (within 5 %), given the fact that no attempt has been made to improve the all-atom polymer force field parameters obtained directly from the CGenFF[65]. Note that the limitation of the current all-atom force field is discussed below.

3.2 PS-b-PMMA Block Copolymer Melt System

Polystyrene and poly(methyl methacrylate) block copolymers (PS-b-PMMA) are widely used because they can phase separate to form periodic nanostructures[68, 69]. Depending on relative lengths of each block, several morphologies such as spherical, cylindrical, gyroid, and lamellar phases can be formed[70, 71]. Such nanostructures could potentially be used for nanoscale templating and separations. For styrene (S) and methyl methacrylate (MMA) of PS-b-PMMA, the number of monomers in each CG bead was calculated to be 4.9 S and 6.9 MMA monomers. For $\epsilon_{S,S}$ and $\epsilon_{MMA,MMA}$, we calculated T_c for each bead based on the group contribution theory, yielding $T_{c,S} = 700$ K and $T_{c,MMA} = 684$ K, which are equivalent to $\epsilon_{S,S} = 1.39$ kcal/mol and $\epsilon_{MMA,MMA} = 1.36$ kcal/mol, respectively. The solubility parameter difference $\Delta\delta$ between S and MMA beads is 3.16 kcal/mol, which corresponds to $\chi = 0.87$ at 400 K ($T^* = 0.57$). $\epsilon_{S,MMA}$ was

determined to be 1.24 kcal/mol from our machine learning method (see METHODS). Note that the conventional Lorentz-Berthelot mixing rule yields $\epsilon_{S,MMA} = (\epsilon_{S,S} \cdot \epsilon_{MMA,MMA})^{1/2}$ = 1.37 kcal/mol, meaning that the system would not microphase separate because $\epsilon_{S,S}$ = 1.39 kcal/mol and $\epsilon_{MMA,MMA}$ = 1.36 kcal/mol.

Six block copolymer structures ($S_{40}MMA_{380}$, $S_{60}MMA_{360}$, $S_{90}MMA_{320}$, $S_{120}MMA_{290}$, $S_{150}MMA_{250}$, and $S_{180}MMA_{220}$) were built with different S volume fractions (f_S), and melt systems with 100 polymer chains in a box of ~20 × 20 × 20 nm^3 were simulated to investigate phase behavior according to various χN. As shown in **Figure 7**, a good agreement with the results from self-consistent field theory (SCFT) calculations[70] demonstrates the validity of our modeling and simulation protocol in *Polymer Builder*. Furthermore, the replacement of CG beads with all-atom monomers allows us to simulate all-atom PS-b-PMMA block copolymer systems that were previously challenging due to limited time and length scales of all-atom simulation. To further investigate the alignment of phase separated domains, we increased the system size two times for P(S_{60}-b-MMA_{360}) and performed CG equilibration for 5 μs. During equilibration, S beads were agglomerated with each other to form spherical phases, and they were packed in a body centered cubic fashion (**Figure S5**).

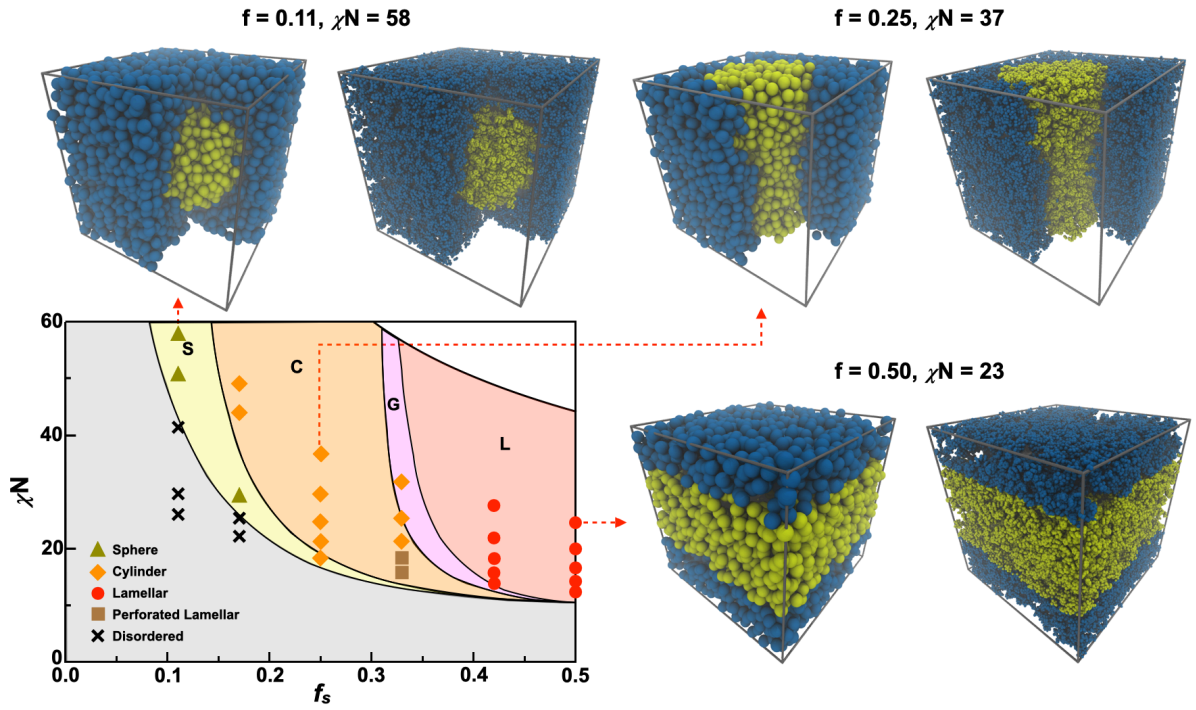

Figure 7. Phase diagram of PS-b-PMMA and snapshots of CG and corresponding all-atom models from *Polymer Builder*, where χN was calculated from the solubility parameter difference using Equation (3) and f_S is the volume fraction of PS. The solid lines represent the boundary between lamellar (L), gyroid (G), cylindrical (C), and spherical (S) phases based on a self-consistent field theory (SCFT) model[70]. Shown together are the observed phases in our CG simulations (symbols) and a few representative snapshots from CG model and corresponding all-atom model. PS and PMMA are colored in yellow and blue, respectively. A quarter of PMMA is omitted for clarity in the $f_S = 0.11$ and $f_S = 0.25$ systems.

As a web-based cyberinfrastructure, it is important to generate a user's system in a reasonable time. **Figure S6A** shows a phase behavior of $S_{40}MMA_{380}$ system at $T^* = 0.50$ ($\chi N = 58$). As the CG simulation proceeded, S beads were agglomerated with each other, and phase separation was completed after 4-μs simulation. With current computing power in CHARMM-GUI, the 4-μs CG simulation takes about 4 hours and the simulation time increases exponentially as the system size increases. To reduce this equilibration time, one can use the Weeks-Chandler-Andersen (WCA) potential (see Equation (5) in METHODS) with $\lambda = 1$ (fully

repulsive for different bead types), which is widely used to accelerate phase separation[54]. When the purely repulsive potential was applied, phase separation of $S_{40}MMA_{380}$ system took less than 100 ns in the simulation time and 10 minutes in real world. However, this intended acceleration could cause unwanted phase separation in certain systems. To overcome this shortcoming, the standard LJ potential ($\lambda = 0$) was applied to the system after accelerated equilibration with the WCA potential. **Figure S6B** shows time series of normalized contact numbers between S beads and corresponding morphologies at various temperature ranging from 0.50 to 1.01. At $T^* = 0.50$ ($\chi N = 58$), spherical phase was maintained during 1-μs equilibration. At $T^* = 1.01$ ($\chi N = 25$), the number of contacts between S beads abruptly decreased during 100 ns and turn into disordered phase. All morphologies obtained by sequentially applying WCA for 100 ns and LJ potential for 100 ns show good match with morphologies in the phase diagram shown in **Figure 7**.

3.3 PEO-b-PEE Amphiphilic Block Copolymer Solution System

We investigated whether the solution system building protocol in *Polymer Builder* could self-assemble important morphologies known to form in polyethylene oxide and polyethyl ethylene block copolymer (PEO-b-PEE)[14, 72]. Three amphiphilic diblock copolymers ($EO_{21}EE_{37}$, $EO_{50}EE_{37}$, and $EO_{92}EE_{37}$) with hydrophilic mass fractions (f_{phil}) of 0.31, 0.51, and 0.66 were built, and 120 $EO_{21}EE_{37}$, 58 $EO_{50}EE_{37}$, and 30 $EO_{92}EE_{37}$ CG polymer chains were randomly distributed with water beads in a box of $10 \times 10 \times 10$ nm^3 (for $EO_{21}EE_{37}$) or a box of $12 \times 12 \times 12$ nm^3 (for $EO_{50}EE_{37}$ and $EO_{92}EE_{37}$). **Figure 8A-F** depicts the final snapshots of CG models and corresponding all-atom structures. Bilayer, cylindrical micelle, and spherical micelle formations are observed at low f_{phil} (< 45 %), intermediate f_{phil} (≈ 50 %), and high f_{phil} (> 60 %), respectively. For a quantitative analysis of bilayers as a function of M_w, five M_w systems ($EO_{10}EE_9$, $EO_{19}EE_{18}$, $EO_{29}EE_{28}$, $EO_{40}EE_{37}$, and $EO_{63}EE_{60}$) with a constant f_{phil} (~43%) were constructed. The density profiles of hydrophobic segments in all systems are shown in **Figure 8G.** Note that the density profile of the smallest M_w copolymer (i.e., $EO_{10}EE_9$) is similar to that of a common phospholipid bilayer. The density

dip at the mid-plane becomes smoothed as M_w increases, indicating that interdigitation and entanglement of hydrophobic chains as the chain length increase. **Figure 8H** shows a quantitative comparison of a logarithmic relationship between the hydrophobic molecular weight ($M_{w,phob}$) and the hydrophobic thickness with theoretical and experimental measurements. An exponent of 0.49 in our results is close to an exponent of 0.50 that is typical of three-dimensional hydrophobic core melts[14]. Therefore, our results demonstrate that *Polymer Builder* provides new ways to analyze novel properties of high M_W copolymer membranes at the atomic resolution and to gain new insight into complex polymers.

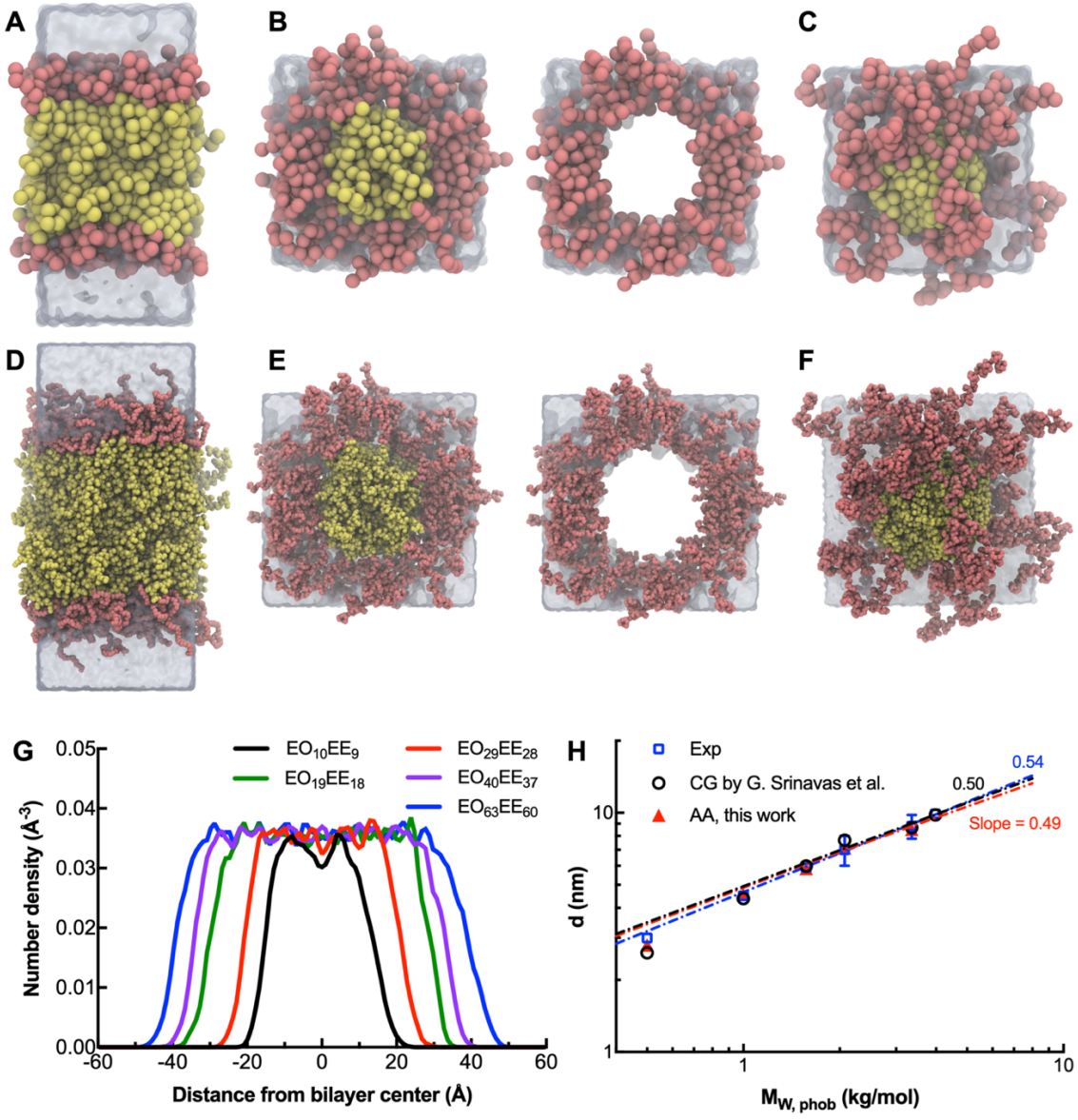

Figure 8. (A-C) Snapshots taken from CG simulations in *Polymer Builder* and (D-E) corresponding all-atom models of PEO-b-PEE in water. (A, D) A bilayer assembly of $EO_{21}EE_{37}$. (B, E) A cylindrical or worm-like micelle assembly of $EO_{50}EE_{37}$ with (left) and without (right) the hydrophobic EE core; n.b., the cylindrical core extends through the periodic boundary conditions. (C, F) A spherical micelle assembly of $EO_{92}EE_{37}$. EO and EE are colored in red and yellow, respectively. (G) Density profiles of hydrophobic blocks in five different PEO-b-PEE bilayer systems with a constant hydrophilic fraction and variable M_w. (H) Scaling of the hydrophobic thickness with the hydrophobic molecular weight ($M_{w,phob}$).

4. LIMITATION AND FUTURE DIRECTIONS

While we have demonstrated that CHARMM-GUI *Polymer Builder* is robust for practical use of modeling and simulation of complex polymer systems, there are some limitations to be noted. As we described above, a pair interaction parameter in the *Polymer Builder* CG model is dependent on χ that is obtained from the solubility parameter differences between Kuhn segments (Equation (3)). Since most group contribution methods for solubility parameters use neutral fragments, the direct prediction of solubility parameters of highly charged polymers such as polyelectrolytes remains challenging. Currently, *Polymer Builder* provides one charged monomer unit (polyacrylic acid). Its solubility parameters are obtained from its neutral form and used to calculate interaction strength in the CG model. After CG equilibration, the system is neutralized by randomly adding counter ions in the all-atom replacement step. Therefore, additional equilibration is necessary to relax such a charged system by letting the ions find their proper positions.

The CG equilibration of most melt or solution systems can be done in a reasonable time in *Polymer Builder*. However, a long equilibrium simulation is required when one tries to perform the simulation at low temperature ($T < T_m$), because the relaxation time is heavily affected by temperature and polymer chain length (see RESULTS AND DISCUSSION). Furthermore, if polymers with long chain lengths or strong intermolecular interactions are suddenly quenched, structural defects can be formed. To avoid such defect

formation, a user needs to perform additional simulated annealing for polymers that have high T_m. *Polymer Builder* provides input files for the simulated annealing simulation with CHARMM, GROMACS, NAMD, LAMMPS, and OpenMM.

Currently, *Polymer Builder* supports a polymer mixture containing up to 99 unique polymer chains, but if a user wants to build a large system of random copolymers, this approach is not efficient, as all unique polymer chains have to be modeled manually. We plan to provide a new option in the system setup step (step 2 in **Figure 1**) to make it easy to build systems containing many unique disordered chains. Furthermore, *Polymer Builder* currently supports 12 single types of solvent. Although one can make some solvents from polymer structure (e.g., octane as 4-mers polyethylene), this approach is limited to cover various solvents. We plan to add more solvents and provide a solvent mixing option in the future update.

In the *Polymer Builder* workflow, adding a new monomer unit requires three major tasks: i) parameterization of a monomer unit (i.e., bonded and nonbonded parameters), ii) generation of patch information (i.e., atom type and partial charge modifications when a monomer unit is connected to other monomers), and iii) parameterization of its CG model (i.e., solubility parameters and Kuhn length of the monomer unit for equilibration). While *Polymer Builder* currently supports a variety of monomer units (more than 60) far more than other available software, we will expand available monomer units by generating the necessary monomer libraries based on the aforementioned three tasks. In addition, we will incorporate a method of building non-linear polymer structures such as branched (star, comb, and brush type) and conjugated polymer units in the future.

The all-atom force field parameters of *Polymer Builder* are based on the CGenFF. Although, the long range structural and bulk properties are well reproduced in most of the currently supported homopolymers, the densities of the polymers containing the ester group show deviations of about 10 % compared to the experimental values. This indicates that there is room for improvement in the current all-atom polymer force field.

5. CONCLUSIONS

We have described a generalized and automated process to build a relaxed polymer system interactively and easily using *Polymer Builder* in CHARMM-GUI. Its versatile and efficient modeling methods to build polymer structures are illustrated by building homopolymers, block copolymers, and random copolymers. The significance of this work is that *Polymer Builder* not only provides single polymer chain modeling capability, but also has carefully built-in methods to generate realistic polymer melt and solution systems using the *Polymer Builder* CG model and its unique all-atom replacement method. The parameterization of the *Polymer Builder* CG model is generalized and extensively validated with various experimental data and all-atom simulations. Using four polymer systems (i.e., PE_{50} melt system, 34 homopolymer melt systems, phase behavior of PS-b-PMMA, and self-assembly of PEO-b-PEE in water), we have also illustrated that the generated systems through *Polymer Builder* are reliable. Together with other CHARMM-GUI modules,[73-79] one can easily study polymer interactions with biological systems such as proteins, nucleic acids, carbohydrates, lipids, detergents, and small molecules. We hope that *Polymer Builder* is useful to carry out innovative and novel polymer modeling and simulation research to acquire insight into structures, dynamics, and underlying mechanisms of complex polymer-containing systems.

Supporting Information

Figure S1. Evaluation of characteristic ratio for all-atom models. Table S1. Flory's characteristic ratio, the number of monomer units in a homopolymer Kuhn segment, and calculated solubility parameter. Figure S2. Evaluation of characteristic ratio for CG models. Table S2. Flory Huggins interaction parameters and corresponding Flory exponents. Figure S3. End-to-end vector autocorrelation functions for CG models with different chain lengths at T* = 0.5, T*= 0.6, and T* = 0.7. Figure S4. Highly entangled polymer system generated from *Polymer Builder*. Figure S5. Microphase separation of PS-b-PMMA and its packing in a

body centered cubic fashion. Figure S6. Acceleration of phase separation using WCA potential.

Supporting Information *Polymer Builder* Tutorial

Tutorial 1. Three single polymer chain examples. Tutorial 2. A solution system of PEO_{76}-PPO_{29}-PEO_{76} in water. Tutorial 3. A ternary blend system of block copolymer (PS_{180}-b-$PMMA_{220}$) and homopolymer (PS_{180} and $PMMA_{220}$). Tutorial 4. Building complex system with PET_{50} and CO_2.

ACKNOWLEDGMENT

This work was supported in part by grants from NSF OAC-193134 and NIH GM138472.

CONFLICT OF INTEREST

The authors declare no competing financial interest.

REFERENCES

1. Choi, Y. K.; Kim, H. J.; Kim, S. R.; Cho, Y. M.; Ahn, D. J., Enhanced thermal stability of polyaniline with polymerizable dopants. *Macromolecules* **2017,** *50*, 3164-3170.
2. Choi, Y. K.; Lee, S. Y.; Ahn, D. J., Hyperconjugation-induced chromism in linear responsive polymers. *J. Mater. Chem. C* **2019,** *7*, 13130-13138.
3. Melling, D.; Martinez, J. G.; Jager, E. W., Conjugated polymer actuators and devices: progress and opportunities. *Adv. Mater.* **2019,** *31*, 1808210.
4. Choi, Y. K.; Lee, D.; Lee, S. Y.; Shin, T. J.; Park, J.; Ahn, D. J., Conjugated polymer nanoparticles in aqueous media by assembly with phospholipids via dense alkyl chain packing. *Macromolecules* **2017,** *50*, 6935-6944.
5. Zhang, P.; Wang, Y.; Lian, J.; Shen, Q.; Wang, C.; Ma, B.; Zhang, Y.; Xu, T.; Li, J.; Shao, Y., Engineering the surface of smart nanocarriers using a pH-/thermal-/GSH-responsive polymer zipper for precise tumor targeting therapy in vivo. *Adv. Mater.* **2017,** *29*, 1702311.
6. Chun, M. J.; Choi, Y. K.; Ahn, D. J., Formation of nanopores in DiynePC–DPPC complex lipid bilayers triggered by on-demand photo-polymerization. *RSC Adv.* **2018,** *8*, 27988-27994.
7. Abolhasani, M. M.; Naebe, M.; Hassanpour Amiri, M.; Shirvanimoghaddam, K.; Anwar, S.; Michels, J. J.; Asadi, K., Hierarchically Structured Porous Piezoelectric Polymer Nanofibers for Energy Harvesting. *Adv. Sci.* **2020**, 2000517.
8. Stuart, M. A. C.; Huck, W. T.; Genzer, J.; Müller, M.; Ober, C.; Stamm, M.; Sukhorukov, G. B.; Szleifer, I.; Tsukruk, V. V.; Urban, M., Emerging applications of stimuli-responsive polymer materials. *Nat. Mater.* **2010,** *9*, 101-113.
9. Gao, S.; Tang, G.; Hua, D.; Xiong, R.; Han, J.; Jiang, S.; Zhang, Q.; Huang, C., Stimuli-responsive bio-based polymeric systems and their applications. *J. Mater. Chem. B* **2019,** *7*, 709-729.
10. Jager, E. W.; Smela, E.; Inganäs, O., Microfabricating conjugated polymer actuators. *Science* **2000,**

290, 1540-1545.

11. Chen, J.; Wu, J.; Qi, J.; Wang, H., Systematic Study of Thermal and (Bio) Degradable Properties of Semiaromatic Copolyesters Based on Naturally Occurring Isosorbide. *ACS Sustainable Chem. Eng.* **2018,** *7*, 1061-1071.
12. Yuan, J.; Zhang, Y.; Zhou, L.; Zhang, G.; Yip, H.-L.; Lau, T.-K.; Lu, X.; Zhu, C.; Peng, H.; Johnson, P. A., Single-junction organic solar cell with over 15% efficiency using fused-ring acceptor with electron-deficient core. *Joule* **2019,** *3*, 1140-1151.
13. Borodin, O.; Smith, G. D., Molecular dynamics simulations of comb-branched poly (epoxide ether)-based polymer electrolytes. *Macromolecules* **2007,** *40*, 1252-1258.
14. Srinivas, G.; Discher, D. E.; Klein, M. L., Self-assembly and properties of diblock copolymers by coarse-grain molecular dynamics. *Nat. Mater.* **2004,** *3*, 638-644.
15. Gartner III, T. E.; Jayaraman, A., Modeling and simulations of polymers: A Roadmap. *Macromolecules* **2019,** *52*, 755-786.
16. Seo, Y.; Shen, K.-H.; Brown, J. R.; Hall, L. M., Role of solvation on diffusion of ions in diblock copolymers: understanding the molecular weight effect through modeling. *J. Am. Chem. Soc.* **2019,** *141*, 18455-18466.
17. Martin, T. B.; Mongcopa, K. I. S.; Ashkar, R.; Butler, P.; Krishnamoorti, R.; Jayaraman, A., Wetting–dewetting and dispersion–aggregation transitions are distinct for polymer grafted nanoparticles in chemically dissimilar polymer matrix. *J. Am. Chem. Soc.* **2015,** *137*, 10624-10631.
18. Panizon, E.; Bochicchio, D.; Monticelli, L.; Rossi, G., MARTINI coarse-grained models of polyethylene and polypropylene. *J. Phys. Chem. B* **2015,** *119*, 8209-8216.
19. Grunewald, F.; Rossi, G.; de Vries, A. H.; Marrink, S. J.; Monticelli, L., Transferable MARTINI model of poly (ethylene oxide). *J. Phys. Chem. B* **2018,** *122*, 7436-7449.
20. Alessandri, R.; Uusitalo, J. J.; de Vries, A. H.; Havenith, R. W.; Marrink, S. J., Bulk heterojunction morphologies with atomistic resolution from coarse-grain solvent evaporation simulations. *J. Am.*

Chem. Soc. **2017,** *139*, 3697-3705.

21. Rossi, G.; Barnoud, J.; Monticelli, L., Polystyrene nanoparticles perturb lipid membranes. *The journal of physical chemistry letters* **2014,** *5*, 241-246.
22. Rossi, G.; Fuchs, P.; Barnoud, J.; Monticelli, L., A coarse-grained MARTINI model of polyethylene glycol and of polyoxyethylene alkyl ether surfactants. *J. Phys. Chem. B* **2012,** *116*, 14353-14362.
23. Kuo, A.-T.; Okazaki, S.; Shinoda, W., Transferable coarse-grained model for perfluorosulfonic acid polymer membranes. *J. Chem. Phys.* **2017,** *147*, 094904.
24. Pervaje, A. K.; Tilly, J. C.; Inglefield Jr, D. L.; Spontak, R. J.; Khan, S. A.; Santiso, E. E., Modeling polymer glass transition properties from empirical monomer data with the SAFT-γ Mie force field. *Macromolecules* **2018,** *51*, 9526-9537.
25. Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C., The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008,** *128*, 244114.
26. Ramos, J.; Peristeras, L. D.; Theodorou, D. N., Monte Carlo simulation of short chain branched polyolefins in the molten state. *Macromolecules* **2007,** *40*, 9640-9650.
27. Haley, B. P.; Wilson, N.; Li, C.; Arguelles, A.; Jaramillo, E.; Strachan, A., Polymer modeler. **2010**.
28. Abbott, L. J.; Hart, K. E.; Colina, C. M., Polymatic: a generalized simulated polymerization algorithm for amorphous polymers. *Theor. Chem. Acc.* **2013,** *132*, 1334.
29. Fortunato, M. E.; Colina, C. M., pysimm: A python package for simulation of molecular systems. *SoftwareX* **2017,** *6*, 7-12.
30. Module, F., Material Studio 6.0. *Accelrys Inc., San Diego, CA* **2011**.
31. Schrodinger, L., Schrodinger software suite. *New York: Schrödinger, LLC* **2011,** *670*.
32. Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M., PACKMOL: a package for building initial configurations for molecular dynamics simulations. *J. Comput. Chem.* **2009,** *30*, 2157-2164.
33. Jo, S.; Kim, T.; Iyer, V. G.; Im, W., CHARMM-GUI: a web-based graphical user interface for

CHARMM. *J. Comput. Chem.* **2008,** *29*, 1859-1865.

34. Jo, S.; Kim, T.; Im, W., Automated builder and database of protein/membrane complexes for molecular dynamics simulations. *PloS one* **2007,** *2*, e880.

35. Lee, J.; Cheng, X.; Swails, J. M.; Yeom, M. S.; Eastman, P. K.; Lemkul, J. A.; Wei, S.; Buckner, J.; Jeong, J. C.; Qi, Y., CHARMM-GUI input generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM simulations using the CHARMM36 additive force field. *J. Chem. Theory Comput.* **2016,** *12*, 405-413.

36. Brooks, B. R.; Brooks III, C. L.; Mackerell Jr, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S., CHARMM: the biomolecular simulation program. *J. Comput. Chem.* **2009,** *30*, 1545-1614.

37. Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E., GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015,** *1*, 19-25.

38. Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K., Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005,** *26*, 1781-1802.

39. Plimpton, S. *Fast parallel algorithms for short-range molecular dynamics*; Sandia National Labs., Albuquerque, NM (United States): 1993.

40. Case, D. A.; Cheatham III, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz Jr, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J., The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005,** *26*, 1668-1688.

41. Jung, J.; Mori, T.; Kobayashi, C.; Matsunaga, Y.; Yoda, T.; Feig, M.; Sugita, Y., GENESIS: a hybrid-parallel and multi-scale molecular dynamics simulator with enhanced sampling algorithms for biomolecular and cellular simulations. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2015,** *5*, 310-323.

42. Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D., OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comp. Biol.* **2017,** *13*, e1005659.

43. Bowers, K. J.; Chow, D. E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossvary, I.; Moraes, M. A.; Sacerdoti, F. D. In *Scalable algorithms for molecular dynamics simulations on commodity clusters*, SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, IEEE: 2006; pp 43-43.

44. Murat, M.; Grest, G. S.; Kremer, K., Statics and dynamics of symmetric diblock copolymers: A molecular dynamics study. *Macromolecules* **1999,** *32*, 595-609.

45. Grest, G. S.; Lacasse, M. D.; Kremer, K.; Gupta, A. M., Efficient continuum model for simulating polymer blends and copolymers. *J. Chem. Phys.* **1996,** *105*, 10583-10594.

46. Weeks, J., D., Chandler, and HC Andersen,". *J. Chem. Phys* **1971,** *54*, 5237.

47. Mark, J. E., *Physical properties of polymers handbook*. Springer: 2007; Vol. 1076.

48. Thakral, S.; Thakral, N. K., Prediction of drug–polymer miscibility through the use of solubility parameter based Flory–Huggins interaction parameter and the experimental validation: PEG as model polymer. *J. Pharm. Sci.* **2013,** *102*, 2254-2263.

49. Lindvig, T.; Michelsen, M. L.; Kontogeorgis, G. M., A Flory–Huggins model based on the Hansen solubility parameters. *Fluid Phase Equilib.* **2002,** *203*, 247-260.

50. Hansen, C. M., *Hansen solubility parameters: a user's handbook*. CRC press: 2007.

51. Stefanis, E.; Panayiotou, C., Prediction of Hansen solubility parameters with a new group-contribution method. *Int. J. Thermophys.* **2008,** *29*, 568-585.

52. Potoff, J. J.; Panagiotopoulos, A. Z., Critical point and phase behavior of the pure fluid and a Lennard-Jones mixture. *J. Chem. Phys.* **1998,** *109*, 10914-10920.

53. Constantinou, L.; Gani, R., New group contribution method for estimating properties of pure compounds. *AIChE J.* **1994,** *40*, 1697-1710.

54. Chremos, A.; Nikoubashman, A.; Panagiotopoulos, A. Z., Flory-Huggins parameter χ, from binary mixtures of Lennard-Jones particles to block copolymer melts. *J. Chem. Phys.* **2014,** *140*, 054909.

55. Groot, R. D.; Warren, P. B., Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.* **1997,** *107*, 4423-4435.

56. Botu, V.; Ramprasad, R., Learning scheme to predict atomic forces and accelerate materials simulations. *Phys. Rev. B* **2015,** *92*, 094306.

57. Botu, V.; Batra, R.; Chapman, J.; Ramprasad, R., Machine learning force fields: construction, validation, and outlook. *J. Phys. Chem. C* **2017,** *121*, 511-522.

58. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res* **2011,** *12*, 2825-2830.

59. Nikoubashman, A.; Lee, V. E.; Sosa, C.; Prud'homme, R. K.; Priestley, R. D.; Panagiotopoulos, A. Z., Directed assembly of soft colloids through rapid solvent exchange. *ACS nano* **2016,** *10*, 1425-1433.

60. Williams, G.; Watts, D. C., Non-symmetrical dielectric relaxation behaviour arising from a simple empirical decay function. *Trans. Faraday Society* **1970,** *66*, 80-85.

61. Nosé, S.; Klein, M., Constant pressure molecular dynamics for molecular systems. *Mol. Phys.* **1983,** *50*, 1055-1076.

62. Parrinello, M.; Rahman, A., Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.* **1981,** *52*, 7182-7190.

63. Hess, B.; Bekker, H.; Berendsen, H. J.; Fraaije, J. G., LINCS: a linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997,** *18*, 1463-1472.

64. Darden, T.; York, D.; Pedersen, L., Particle mesh Ewald: An N· log (N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993,** *98*, 10089-10092.

65. Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.;

Guvench, O.; Lopes, P.; Vorobyov, I., CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2010,** *31*, 671-690.

66. Wang, C.; Mo, B.; He, Z.; Xie, X.; Zhao, C. X.; Zhang, L.; Shao, Q.; Guo, X.; Wujcik, E. K.; Guo, Z., Hydroxide ions transportation in polynorbornene anion exchange membrane. *Polymer* **2018,** *138*, 363-368.

67. Merinov, B. V.; Goddard III, W. A., Computational modeling of structure and OH-anion diffusion in quaternary ammonium polysulfone hydroxide–Polymer electrolyte for application in electrochemical devices. *J. Membr. Sci.* **2013,** *431*, 79-85.

68. Bang, J.; Jeong, U.; Ryu, D. Y.; Russell, T. P.; Hawker, C. J., Block copolymer nanolithography: translation of molecular level control to nanoscale patterns. *Adv. Mater.* **2009,** *21*, 4769-4792.

69. Jin, C.; Olsen, B. C.; Luber, E. J.; Buriak, J. M., Nanopatterning via solvent vapor annealing of block copolymer thin films. *Chem. Mater.* **2017,** *29*, 176-188.

70. Matsen, M. W.; Bates, F. S., Unifying weak-and strong-segregation block copolymer theories. *Macromolecules* **1996,** *29*, 1091-1098.

71. Matsen, M. W.; Bates, F. S., Origins of complex self-assembly in block copolymers. *Macromolecules* **1996,** *29*, 7641-7644.

72. Discher, B. M.; Won, Y.-Y.; Ege, D. S.; Lee, J. C.; Bates, F. S.; Discher, D. E.; Hammer, D. A., Polymersomes: tough vesicles made from diblock copolymers. *Science* **1999,** *284*, 1143-1146.

73. Park, S.-J.; Lee, J.; Qi, Y.; Kern, N. R.; Lee, H. S.; Jo, S.; Joung, I.; Joo, K.; Lee, J.; Im, W., CHARMM-GUI Glycan Modeler for modeling and simulation of carbohydrates and glycoconjugates. *Glycobiology* **2019,** *29*, 320-331.

74. Lee, J.; Patel, D. S.; Ståhle, J.; Park, S.-J.; Kern, N. R.; Kim, S.; Lee, J.; Cheng, X.; Valvano, M. A.; Holst, O., CHARMM-GUI membrane builder for complex biological membrane simulations with glycolipids and lipoglycans. *J. Chem. Theory Comput.* **2018,** *15*, 775-786.

75. Kim, S.; Lee, J.; Jo, S.; Brooks III, C. L.; Lee, H. S.; Im, W., CHARMM-GUI ligand reader and modeler for CHARMM force field generation of small molecules. *J. Comput. Chem.* **2017,** *38*, 1879-1886.

76. Jo, S.; Cheng, X.; Islam, S. M.; Huang, L.; Rui, H.; Zhu, A.; Lee, H. S.; Qi, Y.; Han, W.; Vanommeslaeghe, K., CHARMM-GUI PDB manipulator for advanced modeling and simulations of proteins containing nonstandard residues. In *Advances in protein chemistry and structural biology*, Elsevier: 2014; Vol. 96, pp 235-265.

77. Wu, E. L.; Cheng, X.; Jo, S.; Rui, H.; Song, K. C.; Dávila-Contreras, E. M.; Qi, Y.; Lee, J.; Monje-Galvan, V.; Venable, R. M., CHARMM-GUI membrane builder toward realistic biological membrane simulations. *J. Comput. Chem.* **2014,** *35*, 1997-2004.

78. Park, S.-J.; Lee, J.; Patel, D. S.; Ma, H.; Lee, H. S.; Jo, S.; Im, W., Glycan Reader is improved to recognize most sugar types and chemical modifications in the Protein Data Bank. *Bioinformatics* **2017,** *33*, 3051-3057.

79. Cheng, X.; Jo, S.; Lee, H. S.; Klauda, J. B.; Im, W., CHARMM-GUI Micelle Builder for Pure/Mixed Micelle and Protein/Micelle Complex Systems. *J. Chem. Inf. Model.* **2013,** *53*, 2171-2180.

Table of Contents