polyGraft 1.0: A Program for Molecular Structure and Topology Generation of Polymer-Grafted Hybrid Nanostructures *

Guang Chen†

July 24, 2023

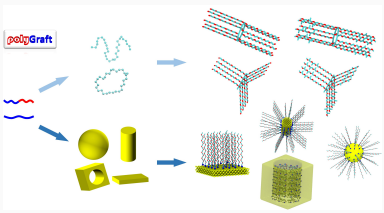
Abstract

Polymer-grafted hybrid materials have been ubiquitously employed in various engineering applications. The design of these hybrid materials with superior performances requires a molecularly detailed understanding of the structure and dynamics of the polymer brushes and their interactions with the grafting substrate. Molecular dynamics simulations are very well suited for the study of these materials which can provide molecular insights into the effects of polymer composition and length, grafting density, substrate composition and curvatures, and nanoconfinement. However, few existing tools are available to generate such systems, which would otherwise reduce the barrier of preparation for such systems to enable high throughput simulations. Here polyGraft, a general, flexible, and easy to use Python program, is introduced for automated generation of molecular structure and topology of polymer grafted hybrid materials for MD simulations purposes, ranging from polymer brushes grafted to hard substrates, to densely grafted bottlebrush polymers. polyGraft is openly accessible on GitHub (https://github.com/nanogchen/polyGraft).

Keywords: Polymer Brush, Hybrid Materials, Nanoconfinement, Nanoparticle, Nanopore

*Manuscript published in J. Comp. Chem. https://doi.org/10.1002/jcc.27206

†Institute of Materials Science, University of Connecticut, Storrs, CT 06269, USA



(75 words.) We develop a Python program, named polyGraft, for the automated generation of molecular structure and topology of polymer-grafted hybrid materials to meet the demand for high-throughput simulations of those systems in which multiple controlling parameters are involved, such as polymer chain length and composition, grafting density, substrate composition and curvatures. A broad range of polymer grafted hard or soft nanostructures can be generated using polyGraft, such as polymer-grafted nanoparticles or nanopores and bottlebrush polymers.

INTRODUCTION

Polymer brushes, formed by grafting polymers to a (organic or inorganic) substrate, are a type of hybrid materials, not only fundamentally important in polymer science, but also practically valued in advanced nanotechnology. Due to their unique properties, they have been widely employed in various applications, such as organic electronic devices[1] where conjugated polymers are used as grafts, and surface engineering, *e.g.*, lubricants[2,3], adhesives[4], and high-performance polymer composites[5]. Remarkably, a broad range of smart materials has been designed based on responsive polymer brushes[6], which has led to the innovation of novel sensors[7], separation membranes[8], and biomedical devices[9], for example, used in drug delivery[10,11], and tissue engineering[12,13], etc.

These hybrid nanomaterials are usually with a soft/soft or hard/soft interface which demonstrates extraordinary properties especially in their assembled structures that are not seen in their bulk due to the confinement effect. Representative examples include, when grafted to a hard material, polymer grafted nanoparticle or spherical brushes[14–16], polymer grafted nanorods[17,18], and polymer grafted nanopores[19,20]. One typical example is poly(ethylene oxide) (PEO) grafted to gold crystal used in biomedical engineering due to its superior biocompatible, optical and nontoxic properties[21,22]. When grafted to a soft material, such as a polymer, the nanostructured material is known as bottlebrush polymer, which is another important polymer brush[23] and can be used for self-healing materials[24], supersoft elastomer[25], and drug delivery vehicle[26].

Molecular dynamics (MD) simulation as one well-established method can probe molecular details of structure and dynamics of these hybrid materials which may not be feasible with experimental or theoretical approaches. With reasonable initial molecular structure and accurate force field, MD simulations can provide new insights into the nanoconfinement effect, soft/hard materials interaction, and hydrophobic/hydrophilic interfaces as well as hydration for which experiments or theories alone are hard to address. Using MD simulations, polymer-grafted hybrid materials can be studied, such as polystyrene-grafted silica nanoparticle[27], poly(L-lactic acid) grafted graphene brush[28], poly(acrylic acid) grafted polyelectrolyte brush[29], and PEO grafted gold planar/spherical/cylindrical brushes[30–33] us-

ing atomistic MD simulations, and polymer grafted nanoparticles using coarse-grained MD simulations[34–36] .

Performing reliable MD simulations of these systems, however, is not trivial. Unlike polymers in solution or in melts, which are a collection of polymers (in melts) or solvated with solvents (in solution) and can be prepared with many existing tools, *e.g.*, PACKMOL[37] or Gromacs modules[38] (*gmx solvate* or *gmx insert-molecules*), the systems for hybrid materials are more difficult to build, as they are an integrated system. Moreover, the chemical space of these hybrid materials is very large since there are usually multiple control parameters involved, such as substrate composition, shape and size, polymer chain length and grafting density. Further consideration of polymer polydispersity and nonuniformly distributed grafting would lead to much more expanded systems for high-throughput investigations. Therefore, a good starting structure is very important, *e.g.*, no close contact of composing atoms especially for hybrid systems with high grafting densities such as polymer-grafted nanopores[33]. Not only the starting structure, but the topology (and corresponding force field parameters) is also very important, which are usually obtained from carefully validated simulation tests against experiments or theories. It is therefore very clear that an MD simulation-oriented program is highly desired for automated generation of the molecular structure and topology of polymer-grafted hybrid materials. Yet, there are very limited existing tools for that end.

Currently, there are a few software/packages that can generate the molecular structure of hybrid materials, yet with limited functionality to be used for polymer-grafted hybrid materials. For example, the doGlycans tool[39] which was specially designed for carbohydrates-based hybrid materials, such as glycolipids, glycoproteins and assembled nanocellulose. However, it is not applicable to generate other synthetic polymer grafted nanostructures since it is based on the building blocks of sugar units defined in GLYCAM[40]. In addition, there is a recent tool named CAT[41] for compound (a ligand) attachment to a crystal lattice, with a couple of examples demonstrated (surfactant grafted cubic lattice, and amino acids grafted metal-organic framework). However, it requires users to define anchoring sites (or grafting points) which would take more effort in generation especially for high grafting density cases while this information is less important than, *e.g.*, the grafting density. Furthermore, it only

produces molecular structures (coordinates) but no topology which is, however, necessary for MD simulations. Even though one can use available software to guess the bonding information, the performance or the accuracy is in question. Lastly, the ligand is processed using the SMILES form[42], and the generated conformation of the ligand tends to be increasingly less optimal for long polymer grafts and at high grafting densities, due to the drastically decreased available spaces, and thus there is a high tendency to deteriorate the optimization of the structure and lead to overlapped structures.

To contribute to filling this gap, we present in this work polyGraft, a Python program for molecular structure and topology generation of polymer grafted nanostructures, ranging from polymer grafted hard materials to densely grafted bottlebrush polymers, as shown in Figure 1. It's designed to lower the entry barrier especially for experimentalists to leverage MD simulations to interpret their results and/or test new hypotheses. Generating those systems requires only general information as inputs (rather than a detailed anchoring process), such as grafting density and the radius of a nanoparticle as well as its element type, *e.g.*, for spherical brushes. Importantly, we also demonstrate equilibrated nanostructures as representative examples by MD simulations to show that the program was MD simulation oriented by design.

METHODOLOGY

Structure of the Code A polymer-grafted hybrid material can be decomposed into two basic components: polymer grafts and the hard/soft substrate, which is the core structure of polyGraft, as shown in Figure 2. The grafting procedure is to graft a polymer (homopolymer, block copolymer or any type) to a substrate satisfying specific grafting density. For polymer grafted hard nanostructure, a polymer as grafts and a hard crystal lattice are required, which can be generated by, *e.g.*, Avogadro[44] and Atomsk[45], respectively. While for polymer-grafted soft nanostructure (bottlebrush polymers), one can use Avogadro to generate the pdb files for both grafts and the backbone. Alternatively for polymers, we have developed an internal generation code for bottlebrush polymer generation considering the cyclic backbone topology. For molecular data processing (*e.g.*, pdb and itp files IO), we employed MDAnalysis[46]. With

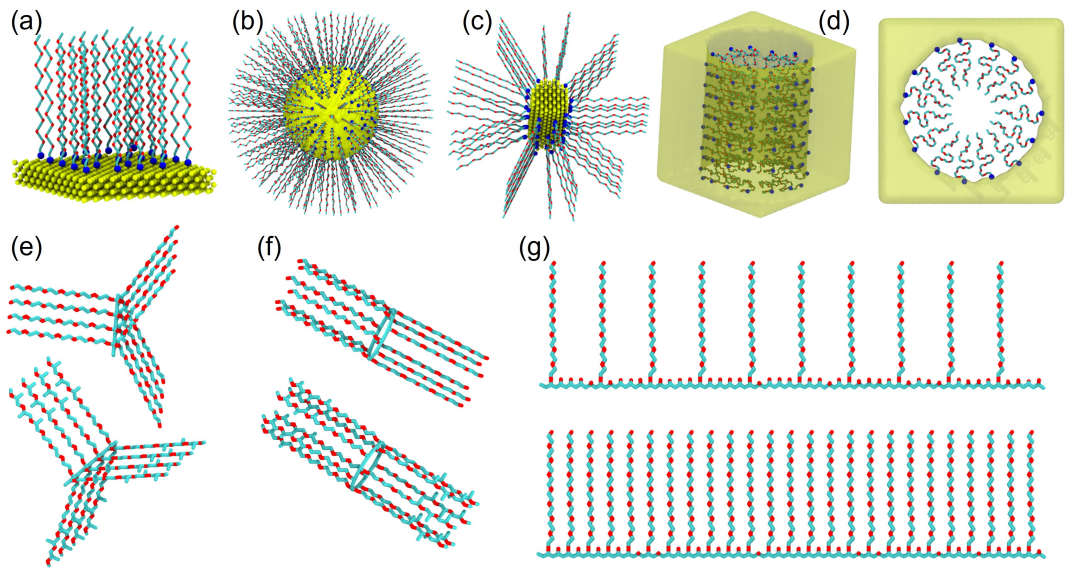


Figure 1: Representative examples of polymer-grafted hybrid nanostructures generated by the polyGraft program. (a) Polymer-grafted planar brush. (b) Polymer-grafted spherical brush. (c) Polymer-grafted nanorod brush. (d) Polymer-grafted nanopore brush in side and top view. (e) Linear bottlebrush polymer with homopolymer (top) and block copolymer grafts (bottom). (f) Cyclic bottlebrush polymer with homopolymer (top) and block copolymer grafts (bottom). (g) Linear bottlebrush polymer with different grafting densities: 0.2 (top) and 0.5 (bottom) grafts/monomer. The snapshots were produced using VMD[43].

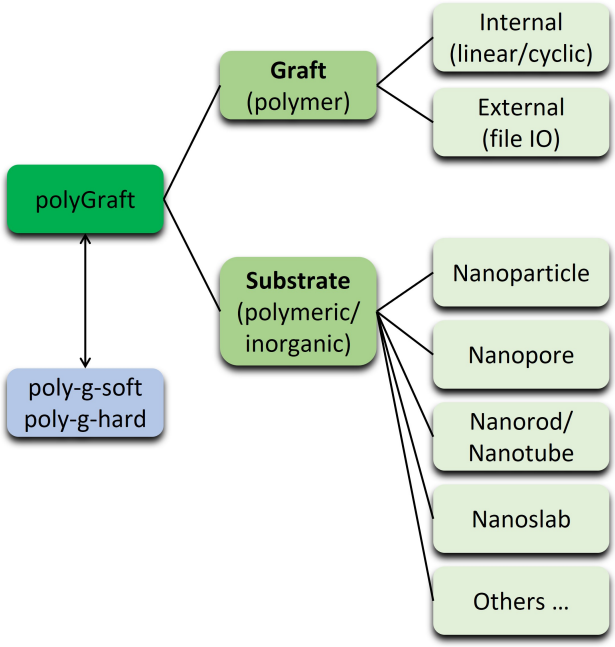


Figure 2: Design of the polyGraft program, based on two basic elements: a graft and a substrate.

Algorithm 1 Grafting scheme for polymer-grafted hybrid materials

procedure GRAFTING
    Generate the lattice of a hard substrate, *e.g.*, a gold nanoparticle; or a polymer as a soft substrate (for bottlebrush polymers)
    Determine the positions of grafting points based on the grafting density
    Loop over each grafting point
    for each grafting point **do**
        `find the normal vector at the grafting point`
        `find the rotation and translation matrix, e.g., Eqs.` (4) - (5)
        `transform the reference graft to the desired place`
    end for
    Save the coordinates of the system
    Generate and save the corresponding topology
end procedure

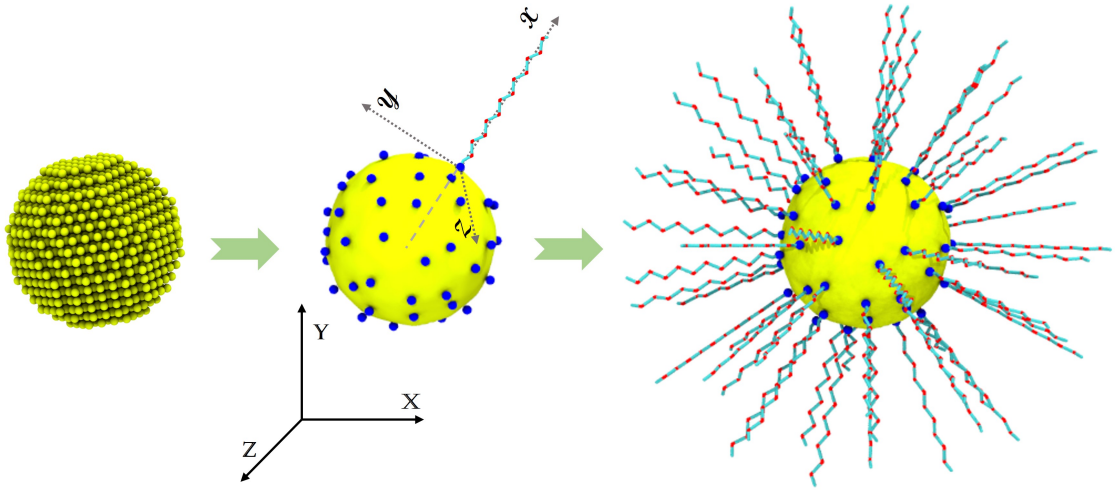


Figure 3: Schematic of the grafting procedure. A polymer (repeat units = 12) grafted to a nanoparticle with 2.0 nm radii at a grafting density of 1.20 nm^{-2}. Substrate atoms are shown in yellow, polymer anchoring atoms shown in blue, other polymer atoms shown in cyan and red.

these basic elements, polyGraft is then applied to assemble them together to produce the desired nanostructure, as summarized in Algorithm 1.

To generate the position of the hybrid nanostructure, which is essentially a collection of the spatial coordinates of all grafts and the substrate, coordinate transformation (translation and rotation) must be carried out on a reference polymer graft. For the topology generation of the hybrid nanostructures, apart from the topology of the graft and substrate, the cross terms between them need to be added, for example, bonds between grafting atoms of the grafts and the substrate atoms to be grafted.

To what follows, a bold letter is used for a vector or a matrix while a regular letter for scalar quantities. A lowercase letter is used for the local coordinate system while an uppercase letter is for the global coordinate system, as shown in Figure 3.

Coordinate translation We start with polymer-grafted planar brush, as shown in Figure 1a, for simplicity yet without loss of generality. In this case, all grafts have the same orientation (with a normal vector $\mathbf{N}$ pointing out of the normal direction of the planar substrate),

therefore coordinate rotation is not needed, assuming their orientation is the same as the reference polymer's otherwise coordinate rotation is needed (see below). Let the length of the planar substrate be d_l, width d_w and depth d_z, and the reference polymer is located along the z-axis and the anchor atom is placed at the origin. The grafting points $\mathbf{X}_{gt}$ can be found as (assume grafting on top of the substrate):

$$\begin{aligned} m &\leftarrow 0, 1, \cdots, \mathcal{N}_l - 1 \\ n &\leftarrow 0, 1, \cdots, \mathcal{N}_w - 1 \\ i &= m + n * \mathcal{N}_l \\ \mathbf{X}_{gt}^T = [x(i), y(i), z(i)] &= [(m + \frac{1}{2})\Delta, (n + \frac{1}{2})\Delta, d_z] \end{aligned} \tag{1}$$

where the subscript T denotes vector transpose. $\mathcal{N}_l = d_{\mathrm{l}}/\Delta$ and $\mathcal{N}_w = d_{\mathrm{w}}/\Delta$ are the number of grafting points in the length and width direction. The total number of grafts is then $\mathcal{N} = \mathcal{N}_l \mathcal{N}_w$. The spacing Δ between neighboring grafting points is calculated by:

$$\Delta = \sqrt{\frac{1}{\sigma}} \tag{2}$$

where σ is the desired grafting density.

The coordinates of all grafts can be found accordingly. Namely, the position of an atom of the polymer in the local coordinate system $\mathbf{x}$ can be updated to the global coordinate system $\mathbf{X}$ by:

$$\mathbf{X} = \mathbf{x} + \mathbf{X}_{gt} + l_g \mathbf{N} \tag{3}$$

where l_g is the bond length between the anchoring atom and the corresponding substrate atom.

Coordinate rotation For surfaces with curvatures, such as nanoparticles, coordinate rotation is also necessary in addition to coordinate translation in order to place a grafting polymer pointing out of the normal direction locally. Let the orientational norm vector of the reference polymer be $\mathbf{n}$ and $\mathbf{N}$ in the local and global coordinate system. The position of a graft in the global coordinate system can be obtained by vector rotation and translation, *i.e.*, rotated to the same direction as the normal vector pointing out of the grafting surface and then translated to the grafting point.

Taking polymer-grafted gold nanoparticles as an example, the procedure is illustrated in Figure 3. The rotation matrix $\mathbf{R}$ can be constructed by the Euler-Rodrigues rotation formula[47] such that $\mathbf{N} = \mathbf{R}\mathbf{n}$:

$$\mathbf{R} = \mathbf{I} + \sin\theta\mathbf{K} + (1 - \cos\theta)\mathbf{K}^2 \tag{4}$$

where $\mathbf{K}$ is the skew-symmetric matrix of the unit vector $\mathbf{k}$ of the rotation axis satisfying $\mathbf{K}\mathbf{n} = \mathbf{k} \times \mathbf{n}$, and θ is the rotation angle. The rotation axis is selected perpendicular to both $\mathbf{n}$ and $\mathbf{N}$, and the rotation angle is the angle between $\mathbf{n}$ and $\mathbf{N}$.

The translation vector $\mathbf{T}$ is formulated, in the example of polymer-grafted gold nanoparticle, as:

$$\mathbf{T} = (R + l_g)\mathbf{N} \tag{5}$$

where R is the radius of the nanoparticle.

With the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$, the position of an atom of the polymer in the local coordinate system $\mathbf{x}$ can be updated to the global coordinate system $\mathbf{X}$ by:

$$\mathbf{X} = \mathbf{R}\mathbf{x} + (R + l_g)\mathbf{N} + \mathbf{X}_C \tag{6}$$

where $\mathbf{X}_C$ is the position of the geometrical center of the nanoparticle.

The coordinate transformation procedure is the same for all other grafts by finding the positions of corresponding grafting points (such that the normal vector $\mathbf{N}$ can be obtained), which can be determined as follows, using still the polymer-grafted nanoparticle as the example. Based on the nanoparticle radius R and the grafting density σ, one gets the surface area of the sphere as $A = 4\pi R^2$, and further the number of grafts $\mathcal{N} = A\sigma = 4\pi R^2\sigma$. The positions of all grafting points can be determined by the Fibonacci lattice[48] to achieve homogeneously grafting on the nanoparticle surface (assuming the center of the nanoparticle is located at the origin, otherwise coordinate translation has to be applied):

$$
\begin{aligned}
i &\leftarrow 0, 1, \cdots, \mathcal{N}-1 \\
\phi &= \arccos(1 - \frac{2i+1}{\mathcal{N}}) \\
\theta &= \frac{2\pi i}{(\sqrt{5}+1)/2} \\
x(i), y(i), z(i) &= R\cos(\theta)\sin(\phi), R\sin(\theta)\sin(\phi), R\cos(\phi)
\end{aligned}
\tag{7}
$$

While for a cylindrical brush (exterior of a nanorod or interior of a nanopore), the procedure is similar to a planar brush (the cylindrical grafting surface can be expanded into a plane with length d_l and width d_w). Assume the axial direction of the cylinder is along the z-axis, then one obtains $d_l = 2\pi R$ and $d_w = l_z$ where R is the radius and l_z is the depth of the cylinder. Correspondingly, the positions of grafting points can be determined as:

$$
\begin{aligned}
m &\leftarrow 0, 1, \cdots, \mathcal{N}_l - 1 \\
n &\leftarrow 0, 1, \cdots, \mathcal{N}_w - 1 \\
i &= m + n * \mathcal{N}_l \\
\theta &= 2\pi/\mathcal{N}_l * m \\
x(i), y(i), z(i) &= R\cos\theta, R\sin\theta, (n + \frac{1}{2})\Delta
\end{aligned}
\tag{8}
$$

with the grafting points, the positions of all grafts can be found accordingly by coordinate translation and rotation, similar to Eq. 6.

Topology generation The topology of the hybrid material contains all grafts and the substrate, which can be determined as follows. Let the index of the reference polymer be $\{1, 2, \cdots, N\}$ (with N being the number of atoms of a polymer graft) and the number of grafts be $\mathcal{N}$. The index of all polymers is thus $graft = \{1, 2, \cdots, N\mathcal{N}\}$. Assume there are N_h atoms of the hard substrate, the index of which is then $sub = \{N\mathcal{N}+1, N\mathcal{N}+2, \cdots, N\mathcal{N}+N_h\}$. For simplicity yet without loss of generality, the bonding information

between polymer anchoring atoms and the hard substrate is then:

$$\begin{matrix} 1 & N_{s1} \\ 1+N & N_{s2} \\ \cdots & \cdots \\ 1+(\mathcal{N}-1)N & N_{s\mathcal{N}} \end{matrix} \tag{9}$$

where N_{si} $(i = 1, 2, \cdots, \mathcal{N})$ is the index of substrate atoms bonded to polymers, which is determined in the grafting procedure (the substrate atom closest to a grafting point).

Other bonded topology information such as angle/dihedral, if necessary, can also be determined similarly. The topology of all other grafts is also very easy to obtain, which is merely a translation of the topology of the reference polymer. For example, the bond between i-th and j-th atom in the reference polymer is translated to $[i, j] \mapsto [i + N * m, j + N * m]$ for m-th graft. The topology of the basic element (graft and hard substrate) can be read from existing files or generated by the program for simple structures. For the hard substrate, only bonds are considered by default, which is formed by finding the nearest neighbors of an atom.

The overall procedure to generate the molecular structure of polymer-grafted soft nanostructure, such as bottlebrush polymers, is pretty similar to that of polymer-grafted hard nanostructure, discussed above. However, a different approach to generate topology is adopted in order to correctly produce angles and dihedrals of the hybrid structure and also to avoid "reinventing the wheel". Specifically, we employ existing tools (Gromacs pdb2gmx[38]) to generate the topology for bottlebrush polymers. As such, one needs only to define a residue file to generate the topology.

RESULTS

Substrate and Grafts The basic structure of polyGraft includes two important elements: a polymer as the grafts and a crystal lattice (or a polymer) as the substrate to be grafted with the grafts. A code snippet is demonstrated for the generation of a gold nanoparticle with a 2.0 nm radii, and a polymer (PEO) are shown in Figure S1 and Figure S2 of the Supporting Information.

```
# import polyGraft
from polyGraft import polyGraft
from polymer import Polymer
from crystal import Crystal

# import peo
peo = Polymer(poly_name="PEO")
peo.readGRO("PEO12_line.gro")
peo.readITP("PEO12.itp")

# import nanoparticle
nanoparticle = Crystal("nanoparticle", "Au", radius=20.0)
nanoparticle.readPDB("AuNP.pdb", guessing_bond=True, lattice_const=4.08)

# graft together based on grafting density (unit in A^-2)
np_g_peo = polyGraft(nanoparticle, peo)
gft_density = 0.0250
np_g_peo.setGraftingDensity(gft_density)

# set substrate atoms to be grafted and generate the grafted structure
peo_g_np.setGftAtoms("Au")
np_g_peo.genGraftStruct()
np_g_peo.toGRO("peo_g_np.gro")
np_g_peo.toITP("peo_g_np.itp")
```

Listing 1: A code snippet for the generation of PEO12 grafted gold nanoparticle with 2.0 nm radii at the grafting density of 2.5 nm^{-2}.

Polymer Grafted to Hard Material With the hard substrate and polymer structure at hand, generating the molecular structure and topology of polymer-grafted hard material is very straightforward. Taking PEO-grafted gold nanoparticle as an example, the code snippet is shown in Listing 1.

In Figure 1a-d, various polymer-grafted hard nanostructures generated by polyGraft are demonstrated, including PEO-grafted planar brush, spherical brush, cylindrical brush outside of a nanorod, and cylindrical brush inside of a nanopore. PEO in two different

conformations (straight and curved) used for grafting were generated using Avogadro. For grafting on planar and convex surfaces, the straight form of PEO was applied; while for grafting on concave surfaces with limited accessible spaces, the curved form was employed to avoid interchain overlap or overlapping with the pore wall, which ensures a good starting molecular structure for MD simulations.

To show the versatility of the polyGraft program, polymer grafted nanoparticles with various nanoparticle sizes and grafting densities were generated using polyGraft, as shown in Figure 4. For simplicity of demonstration, PEO with chain length $N = 12$ is used. Accommodating for chain length variance would be rather simple provided that the polymer with the desired length is generated/read and used in generation. With these initial molecular structures and corresponding topology at hand, the effect of polymer chain length and grafting density when grafted on nanoparticles with different radii can be investigated, which provides molecular insights on the structural and hydration properties for water-soluble polymers like PEO[31,32].

The procedure for generating polymer-grafted hard nanostructures beyond nanoparticles is the same, given the specific lattice structure of the hard substrate. For example, PEO-grafted gold nanopores with different chain lengths and grafting densities grafted on the interior of gold nanopores with various radii have been generated by polyGraft and studied using atomic MD simulations to elucidate the interplay between the three governing parameters on the structural and hydration properties of PEO layers[33].

Polymers Grafted to Soft Material Generating the molecular structures of polymer-grafted soft materials, or bottlebrush polymer, is as simple as in the previous case. Here using the poly(vinyl alcohol) (PVA) as the backbone, and PEO or poly(ethylene oxide)-block-poly(propylene oxide) (PEO-b-PPO) as the grafts for illustration, a wide variety of molecular architectures are generated using polyGraft for demonstration, considering the topology effect of the backbone (linear vs cyclic), the backbone length (N_{bb}), the composition (homopolymer vs block copolymer, *e.g.*, core-shell bottlebrush polymers[49]) and length of the side chain (N_{sc}), and the grafting density, as shown in Figure 1e-g and Figure 5.

MD simulation of these bottlebrush polymers can gain molecular understanding of the

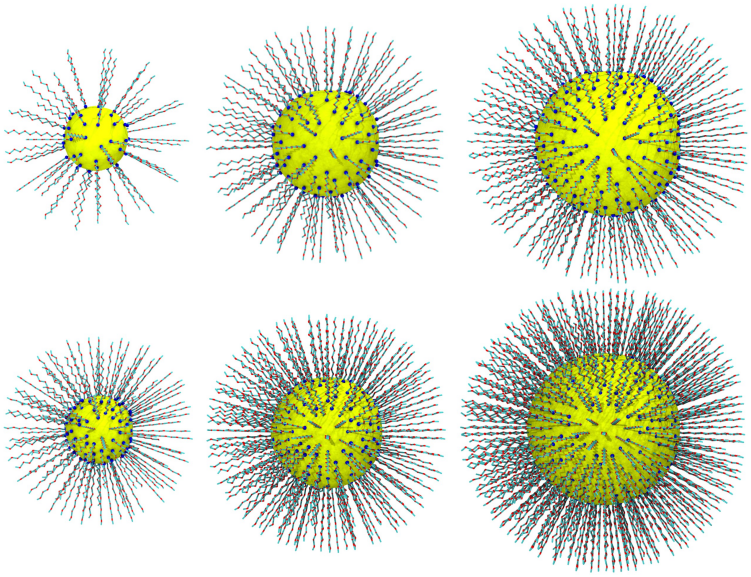


Figure 4: Examples of polymer-grafted gold nanoparticles with various particle sizes (radii of 2, 3, and 4 nm from the left to the right) and grafting densities (1.2 nm^{-2} on the top and 2.5 nm^{-2} on the bottom).

structure property relationships. For example, the backbone gains rigidity upon side chain length increase[23] and the induced steric effect significantly reduces the possibility to form molecular entanglements[25] compared to conventional linear polymers. In addition, the backbone topology and side chain length affect the structural and hydration properties, revealed by atomic MD simulations using the structure and topology generated by polyGraft[50]. Apart from the backbone topology and chain length of backbones and grafts, the grafting density is also an important parameter determining the structural properties of bottlebrush polymers. To accommodate for this factor, polyGraft can generate molecular structures with various grafting densities, as shown in Figure 5d.

Note that since there is no "normal vector" well-defined for a linear polymer substrate (a line), the normal vectors were selected to have 0 or 120 (linear backbone) and 180 degrees (cyclic backbone) spacing between grafts to avoid overlapping. For BBPs with cyclic

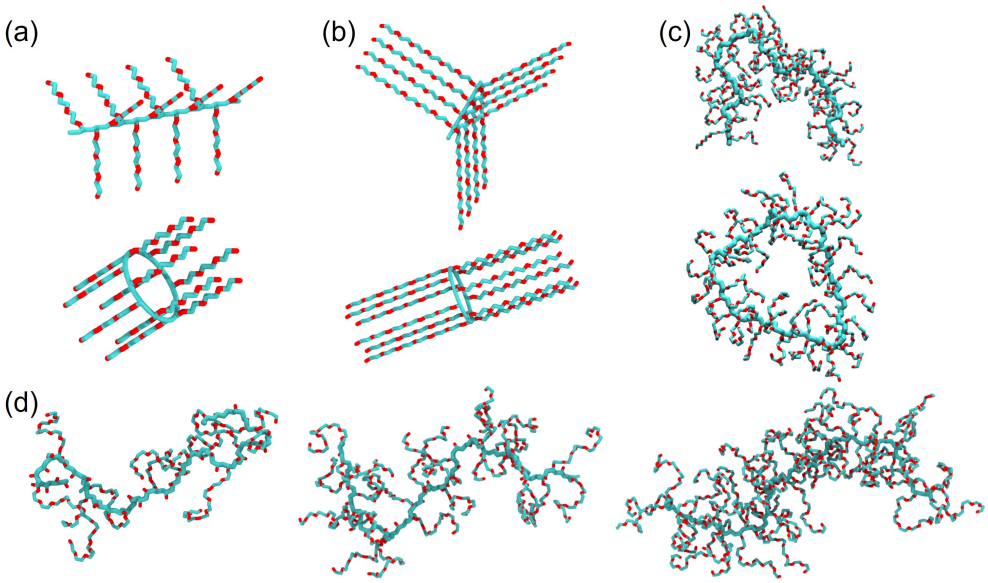


Figure 5: Bottlebrush polymer architectures with different backbone, grafts lengths, and grafting density. (a) Linear l-PVA12-g-PEO3 and cyclic c-PVA12-g-PEO3. (b) Linear l-PVA12-g-PEO6 and cyclic c-PVA12-g-PEO6. (c) Equilibrated structures of linear l-PVA50-g-PEO4 and cyclic c-PVA50-g-PEO4. (d) Equilibrated structures of linear l-PVA50-g-PEO8 with grafting density of 0.2, 0.5, and 1.0 grafts/monomer.

backbones, we use a built-in generation method:

$$\begin{aligned} i &\leftarrow 0, 1, \cdots, 2N_{bb} - 1 \\ \theta &= \pi / N_{bb} * i \\ x(i), y(i), z(i) &= R\cos\theta, R\sin\theta, 0 \end{aligned} \tag{10}$$

where $R = 0.5l/\sin[\pi/(2N_{bb})]$ is the radius of the backbone ring, and l is the carbon-carbon bond length of PVA.

DISCUSSION AND CONCLUSIONS

In summary, we have developed a Python program (polyGraft) for molecular structure and topology generation of polymer grafted nanostructures, to meet the demand for high-throughput simulations of polymer grafted hybrid materials when multiple controlling parameters are involved. polyGraft was developed based on two basic elements: substrates

(a hard lattice crystal or a soft polymer) and grafts (usually a polymer). Despite this simple design, it can generate very diverse polymer grafted hybrid materials, covering a wide range of grafting density, chain length, curvatures, chain composition (homopolymer, block copolymer, or any type), and topology effect (linear vs cyclic in bottlebrush polymers). The capability of polyGraft was demonstrated using a variety of examples, including polymer-grafted hard nanostructures and polymer-grafted soft nanostructures. Here its features, potential, and limitations are discussed.

First off, polyGraft can be generalized. It's been developed for the generation of molecular structure and topology; while no assumption of the force field has been made (though we demonstrated some equilibrated structures via MD simulation using the OPLS-AA force field[51]) and thereby not limited to certain force field or MD simulation engine. Therefore, it provides large freedom and we leave it to the users to select the most suitable force field or MD simulation platforms for their simulations. In order to run simulations in other engines, one may need to convert the gro/itp files to what suits the need by, *e.g.*, MDAnalysis[46].

It also has good flexibility and versatility. Although here PEO was used as the grafts, in principle, any type of polymer can be employed, *e.g.*, polysaccharides, polypeptides and DNA strand. In addition, though the atomistic molecular structure and topology is demonstrated, coarse-grained nanostructures (CG) can also be accommodated given the gro/itp files of the CG substrates and grafts, following the same philosophy. Moreover, though the lattice structure of hard substrates composed by single atom type (*i.e.*, gold atom) was used for demonstration, polyGraft is readily capable of grafting polymers to multi atom-type lattice structure (*e.g.*, silicon oxide), if providing only the positions of atoms to be grafted (*e.g.*, silicon atoms) instead of all lattice positions in the grafting procedure. Here in all examples, all grafts are the same (length and composition), while it is also of great research importance to investigate, for example, the bimodal distribution[52] of the molecular weight or multi-component polymers for grafting (*e.g.*, PEO and PPO) in a patchy or Janus form[15]. Still, polyGraft is easily extended to accommodate for distinctive grafts, which can be achieved in the grafting procedure. For instance, as per the objectives of grafting, one can select desired polymers in specific positions, for example, using PEO (or short polymer) for odd indices of grafting points while using PPO (or long polymer) for even indices. The

indices can also be random if randomly mixed grafting is desired. For soft nanoparticles composed of polymers (*e.g.*, polymer micelles), we note that, polyGraft is not yet applicable to generate the topology (though the geometry or the structure is easy to create), except that users provide the topology for the whole system. For the bottlebrush polymer generation, we note that currently polyGraft does not support arbitrary combinations of graft and backbone composition other than PVA-g-PEO. Although the molecular structure of any type of bottlebrush polymers can be generated readily, the topology cannot. Nonetheless, with the PVA-g-PEO example provided, one can accommodate other types following the same procedure (define the residue used to generate topology) with minimal effort. Automated generation for any type of polymers as grafts and backbones might be developed in a future version.

Finally and importantly, polyGraft is completely free and open-sourced, which is openly accessible on GitHub. We welcome extensions and feature request if it is also in alignment with our research interest. We hope it will be useful to researchers with interests in polymer-grafted hybrid materials, especially experimentalists that want to use MD simulations to interpret their results and/or test new hypotheses.

DATA AND SOFTWARE AVAILABILITY

The source code of polyGraft as well as the molecular structure and topology files demonstrated in this paper are available at https://github.com/nanogchen/polyGraft.

ACKNOWLEDGMENTS

This research is supported by the National Science Foundation under Grant No. DMR-1916864 and No. CHE-2004072. Constructive comments and suggestions from Prof. Elena Dormidontova are gratefully acknowledged.

References

1. A. Bousquet, H. Awada, R. C. Hiorns, C. Dagron-Lartigau, and L. Billon, Prog. Polym. Sci. **39**, 1847 (2014).

2. J. Klein, Y. Kamiyama, H. Yoshizawa, J. N. Israelachvili, G. H. Fredrickson, P. Pincus, and L. J. Fetters, Macromolecules **26**, 5552 (1993).

3. J. Klein, E. Kumacheva, D. Mahalu, D. Perahia, and L. J. Fetters, Nature **370**, 634 (1994).

4. M. M. Feldstein, E. E. Dormidontova, and A. R. Khokhlov, Prog. Polym. Sci. **42**, 79 (2015).

5. S. K. Kumar, N. Jouault, B. Benicewicz, and T. Neely, Macromolecules **46**, 3199 (2013).

6. S. Peng and B. Bhushan, RSC Adv. **2**, 8557 (2012).

7. N. Ayres, Polym. Chem. **1**, 769 (2010).

8. J. J. Keating IV, J. Imbrogno, and G. Belfort, ACS Appl. Mater. Interfaces **8**, 28383 (2016).

9. M. Krishnamoorthy, S. Hakobyan, M. Ramstedt, and J. E. Gautrot, Chem. Rev. **114**, 10976 (2014).

10. A. K. Bajpai, S. K. Shukla, S. Bhanu, and S. Kankane, Prog. Polym. Sci. **33**, 1088 (2008).

11. C. Feng and X. Huang, Acc. Chem. Res. **51**, 2314 (2018).

12. W. Kim and J. Jung, BMB Rep. **49**, 655 (2016).

13. İ. K. Açarı, E. Sel, İ. Özcan, B. Ateş, S. Köytepe, and V. K. Thakur, Adv. Colloid Interface Sci. p. 102694 (2022).

14. R. Sardar, A. M. Funston, P. Mulvaney, and R. W. Murray, Langmuir **25**, 13840 (2009).

15. A. J. Chancellor, B. T. Seymour, and B. Zhao, Anal. Chem. **91**, 6391 (2019).

16. N.-N. Zhang, X. Shen, K. Liu, Z. Nie, and E. Kumacheva, Acc. Chem. Res. **55**, 1503 (2022).

17. H. Chen, L. Shao, Q. Li, and J. Wang, Chem. Soc. Rev. **42**, 2679 (2013).

18. J. Zheng, X. Cheng, H. Zhang, X. Bai, R. Ai, L. Shao, and J. Wang, Chem. Rev. **121**, 13342 (2021).

19. M. Tagliazucchi and I. Szleifer, Soft Matter **8**, 7292 (2012).

20. T. Ma, J.-M. Janot, and S. Balme, Small Methods **4**, 2000366 (2020).

21. C. Cruje and D. Chithrani, J. Nanomed. Res. **1** (2014).

22. N. Verimli, A. Demiral, H. Yılmaz, M. Çulha, and S. S. Erdem, Appl. Biochem. Biotechnol. **189**, 709 (2019).

23. R. Verduzco, X. Li, S. L. Pesek, and G. E. Stein, Chem. Soc. Rev. **44**, 2405 (2015).

24. R. Xie, I. Lapkriengkri, N. B. Pramanik, S. Mukherjee, J. R. Blankenship, K. Albanese, H. Wang, M. L. Chabinyc, and C. M. Bates, Macromolecules **55**, 10513 (2022).

25. W. F. Daniel, J. Burdyńska, M. Vatankhah-Varnoosfaderani, K. Matyjaszewski, J. Paturej, M. Rubinstein, A. V. Dobrynin, and S. S. Sheiko, Nat. Mater. **15**, 183 (2016).

26. A. Detappe, H. V.-T. Nguyen, Y. Jiang, M. P. Agius, W. Wang, C. Mathieu, N. K. Su, S. L. Kristufek, D. J. Lundberg, S. Bhagchandani, et al., Nat. Nanotechnol. pp. 1–9 (2023).

27. T. V. Ndoro, E. Voyiatzis, A. Ghanbari, D. N. Theodorou, M. C. Böhm, and F. Müller-Plathe, Macromolecules **44**, 2316 (2011).

28. K. Hasheminejad, A. Montazeri, and H. Hasheminejad, Eur. Polym. J. **148**, 110351 (2021).

29. H. S. Sachar, B. S. Chava, T. H. Pial, and S. Das, Macromolecules **54**, 6342 (2021).

30. U. R. Dahal, Z. Wang, and E. E. Dormidontova, Macromolecules **50**, 6722 (2017).

31. U. Dahal, Z. Wang, and E. E. Dormidontova, Macromolecules **51**, 5950 (2018).

32. U. Dahal and E. E. Dormidontova, Macromolecules **53**, 8160 (2020).

33. G. Chen and E. Dormidontova, Macromolecules **55**, 5222–5232 (2022).

34. J. G. Ethier and L. M. Hall, Macromolecules **51**, 9878 (2018).

35. J. G. Ethier, L. F. Drummy, R. A. Vaia, and L. M. Hall, ACS nano **13**, 12816 (2019).

36. N. K. Hansoge, A. Gupta, H. White, A. Giuntoli, and S. Keten, Macromolecules **54**, 3052 (2021).

37. L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, J. Comput. Chem. **30**, 2157 (2009).

38. D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, J. Comput. Chem. **26**, 1701 (2005).

39. R. Danne, C. Poojari, H. Martinez-Seara, S. Rissanen, F. Lolicato, T. Róg, and I. Vattulainen, J. Chem. Inf. Model. **57**, 2401 (2017).

40. R. Woods, K. Kirschner, A. Yongye, S. Tschampel, J. Gonzalez-Outeirino, et al., J. Comput. Chem. **29**, 622 (2008).

41. B. van Beek, J. Zito, L. Visscher, and I. Infante, J. Chem. Inf. Model. (2022).

42. D. Weininger, J. Chem. Inf. Comput. **28**, 31 (1988).

43. W. Humphrey, A. Dalke, and K. Schulten, J. Mol. Graphics **14**, 33 (1996).

44. M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, J. Cheminf. **4**, 1 (2012).

45. P. Hirel, Comput. Phys. Commun. **197**, 212 (2015).

46. N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, J. Comput. Chem. **32**, 2319 (2011).

47. J. S. Dai, Mech. Mach. Theory **92**, 144 (2015).

48. Á. González, Math. Geosci. **42**, 49 (2010).

49. S. Kripotou, C. Psylla, K. Kyriakos, K. N. Raftopoulos, J. Zhao, G. Zhang, S. Pispas, C. M. Papadakis, and A. Kyritsis, Macromolecules **49**, 5963 (2016).

50. G. Chen and E. Dormidontova, Macromolecules **56**, 3286 (2023).

51. W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

52. A. Rungta, B. Natarajan, T. Neely, D. Dukes, L. S. Schadler, and B. C. Benicewicz, Macromolecules **45**, 9303 (2012).